
Bachelorarbeit

Janine Brettschneider

**Generierung von
Oligonukleotiden zur Identifi-
kation von *Candida*-Arten
durch die Entwicklung eines
darauf angepassten
Algorithmus‘**

Mittweida, 2011

BACHELORARBEIT

Generierung von Oligonukleotiden zur Identifi- kation von *Candida*-Arten durch die Entwicklung eines darauf angepassten Algorithmus‘

Autor:
Frau

Janine Brettschneider

Studiengang:
Biotechnologie/Bioinformatik

Seminargruppe:
Bi08w2-B

Erstprüfer:
Prof. Dr. rer. nat. Dirk Labudde

Zweitprüfer:
Dr. rer. nat. Werner Brabetz

Einreichung:
Mittweida, 22.08.2011

bachelor thesis

Design of oligonucleotides for the identification of several *Candida spp.* with a self- developed algorithm

author:

Ms.

Brettschneider Janine

course of studies:

Biotechnology/Bioinformatics

seminar group:

Bi08w2-B

first examiner:

Prof. Dr. rer. nat. Dirk Labudde

second examiner:

Dr. rer. nat. Werner Brabetz

submission:

Mittweida, 22/08/2011

defence/evaluation:

Mittweida, 2011

Bibliografische Beschreibung:

Brettschneider, Janine:

Generierung von Oligonukleotiden zur Identifikation von *Candida*-Arten durch die Entwicklung eines darauf angepassten Algorithmus' - 2011. – IV-IX, II-III

Mittweida, Hochschule Mittweida, Fakultät Mathematik, Naturwissenschaften, Informatik, Bachelorarbeit, 2011

Referat:

Die vorliegende Arbeit befasst sich mit der Generierung von Primern und Sonden zur Identifikation verschiedener *Candida*-Arten. Hierzu erfolgte zunächst die Evaluierung gebräuchlicher Primer-Programme. Da keines den geforderten Ansprüchen genügt, wurde ein eigener Algorithmus entworfen. Mit Hilfe dessen erfolgte abschließend die Generierung der Primer und Sonden.

Inhalt

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VII
Konventionen in dieser Arbeit.....	VIII
Danksagung.....	IX
1. Einleitung.....	1
2. Grundlagen	3
2.1 <i>DNA-der Träger unserer Vergangenheit.....</i>	3
2.1.1 Allgemeines.....	3
2.1.2 Aufbau.....	3
1.2 <i>Polymerase-Kettenreaktion</i>	5
1.2.1 Allgemeines.....	5
1.2.2 Ablauf.....	6
1.3 <i>Microarray</i>	8
1.3.1 Allgemeines.....	8
1.3.2 DNA-Microarray.....	8
1.4 <i>Strategie zur Identifikation der Pilze</i>	9
3. Oligonukleotide	11
4. Oligonukleotidtools im Internet.....	17
5. AP²D	20
5.1 <i>Klassenhierarchie.....</i>	20
5.2 <i>Sequenz Erfassung</i>	22
5.3 <i>Default-Werte</i>	26
5.4 <i>Primersuche</i>	27
5.4.1 Generierung der Primer.....	27
5.4.2 chemische Eigenschaften der generierten Primer	29
5.4.3 Bewertung der Primer nach ihren chemischen Eigenschaften	31

5.4.5 Bewertung der Primer nach ihrer Neigung zu Fehlhybridisierungen	33
5.4.6 Bewertung der Primerpaare nach allen Eigenschaften	38
5.4.7 Finden des besten Primerpaares.....	38
5.4.8 Spezifitätsüberprüfung.....	40
5.4.9 Auswertung der BLAST-Suche.....	43
5.5 Generierung der Sonden.....	45
5.5.1 Eingabe.....	45
5.5.2 Generierung der Sonden	46
5.5.3 chemische Eigenschaften der generierten Sonden.....	46
5.5.4 Bewertung der Sonden nach ihren chemischen Eigenschaften	47
5.5.5 Neigung der Sonden zu Fehlhybridisierungen	48
5.5.6 Bewertung der Sonden nach allen Eigenschaften	48
5.5.7 Finden der besten Sonde	48
5.5.8 Spezifitätsüberprüfung.....	48
6. Mittels AP²D erzeugte Primer und Sonden	49
6.1 <i>Candida albicans</i>	49
6.2 <i>Candida glabrata</i>	50
6.3 <i>Candida guilliermondii</i>	51
6.4 <i>Candida krusei</i>	52
6.5 <i>Candida parapsilosis</i>	53
6.6 <i>Candida tropicalis</i>	54
7. Zusammenfassung und Ausblick.....	55
Literatur	XI
Internetquellen	XVI
Selbstständigkeitserklärung	XVII

Abbildungsverzeichnis

Abbildung 2-1: historische Bilder der DNA.....	3
Abbildung 2-2: Oligonucleotids aus zwei Nukleotiden [Zvelibil 2008].....	4
Abbildung 2-3: Darstellung der DNA [Zvelibil 2008].....	5
Abbildung 2-4: Schritte der PCR [URL 2].....	7
Abbildung 3-5: Watson-Crick-Basenpaarung [URL 3].....	11
Abbildung 3-6: mögliche Fehlhybridisierungen bei Primern	12
Abbildung 3-7: Darstellung von zwei unterschiedlichen Hybridisierungskurven während einer PCR [Yuryev 2007].....	14
Abbildung 5-8: FASTA-Format des Genbank-Eintrages 188497078	23
Abbildung 5-9: Genbank-Format des Genbank-Eintrage 188497078	23
Abbildung 5-10: Darstellung der Bereiche der eingegebenen Sequenz, die zur Generierung der fwd und rev Primer genutzt werden [Kämpke 2001]	27
Abbildung 5-11: Generierung aller möglichen Primer der Länge 4 anhand einer vorgegebenen Inputsequenz	27
Abbildung 5-12: Beispiel zur Berechnung des SA-Scores	33
Abbildung 5-13: Beispiel zur Berechnung des SEA-Scores.....	35
Abbildung 5-14: Generierung einer PrimerpaarListe	39
Abbildung 5-15: Ausschnitt einer mittels des BLAST-Webservice generierten out.txt.Datei.....	44
Abbildung 6-16: Eclipse-Screenshot zum Primer-Design von C. albicans	49

Tabellenverzeichnis

Tabelle 3-1: allgemeingültige SantaLucia-Werte [SantaLucia 1986].....	15
Tabelle 4-2: Auflistung der potentiell geeigneten Internetprogramme.....	17
Tabelle 4-3: Evaluierung der Internetprogramme	19
Tabelle 5-4: Vergleich von String und StringBuilder	22
Tabelle 5-5: Default-Werte des Programmes.....	26
Tabelle 5-6: DNA Identity Matrix des EBI.....	40
Tabelle 6-7: Primerpaar für <i>Candida albicans</i>	50
Tabelle 6-8: Primerpaar für <i>Candida glabrata</i>	51
Tabelle 6-9: Primerpaar für <i>Candida guilliermondii</i>	52
Tabelle 6-10: Primerpaar für <i>Candida krusei</i>	52
Tabelle 6-11: Primerpaar für <i>Candida parapsilosis</i>	53
Tabelle 6-12: Primerpaar für <i>Candida tropicalis</i>	54

Abkürzungsverzeichnis

A	Adenin
bp	Basenpaare
C	Cytosin
DB	Datenbank
DNA	Desoxribonukleinsäure
dNTP	Desoxyribonukleotidtriphosphat
engl.	englisch
fwd	forward; bei DNA-Sequenzen, die in 5'-3'-Orientierung dargestellt sind
G	Guanin
internat.	international
ITS	Internal transcribed spacer
lat.	Lateinisch
NN	Nearest-Neighbor
PCR	Polymerasekettenreaktion
rDNA	ribosomale DNA
rev	reverse; bei DNA-Sequenzen, die reverse komplementär zur 5'-3'-Orientierung verlaufen
SNP	single nucleotide polymorphism
T	Thymin
Tm	Schmelztemperatur

Konventionen in dieser Arbeit

Um die Lesbarkeit zu erhöhen wurden folgende Konventionen festgelegt:

Kursiv

Fett

Monospace

Monospace

- erstmalige Erwähnung eines Fachbegriffes
- kennzeichnet eine beginnende Erläuterung des fettgeschriebenen Begriffes/der fettgeschriebenen Begriffe
- kennzeichnet das Ergebnis einer Formel
- kennzeichnet folgende Java-Programmelemente: Klassennamen, Methodennamen, Code-Ausschnitte
- kennzeichnet Pseudo-Codes

Danksagung

An erster Stelle möchte ich mich bei Prof. Dr. rer. nat. Dirk Labudde bedanken, da ich erst durch ihn die Möglichkeit bekam, dieses Thema zu bearbeiten und des weiteren möchte ich mich bei ihm dafür bedanken, dass er immer wieder neue Anregungen und Ideen zu meinem Bachelorthema beisteuerte.

Weiterhin möchte ich mich bei Dr. rer. nat. Werner Brabetz von der Firma Biotype Diagnostic GmbH (Dresden) bedanken, durch den die Fragestellung dieses Projektes erst entstand. Die Gespräche mit ihm eröffneten immer neue Möglichkeiten für Weiterentwicklungen und gaben stets einen interessanten Einblick in die Pilzdiagnostik-Thematik.

Ebenso möchte ich mich bei Florian Heinke (B.Sc.) bedanken, der mir intensiv bei der Implementierung des Webservices half und mir durch seinen großen Wissensfundus im bioinformatischen, als auch informatischen Bereich viele Anregungen gab. Er fand auch bei all meinen sonstigen Fragen immer Zeit für mich und die vielen „visionären“ Unterhaltungen mit ihm waren stets sehr motivierend.

Ebenfalls gilt Gabriel Kind ein großes Dankeschön, da er trotz seiner eigenen Bachelorarbeit und der Tatsache, dass er im Urlaub in Montenegro war, immer wieder Zeit fand, sich meinen auftretenden Fehlern zu widmen. Gerade in den Tagen kurz vor Abgabe der Arbeit saß er stets mit mir bis früh in den Morgen nach 4 Uhr um meine Arbeit zu korrigieren und letzte informatische Schwachstellen zu beseitigen. Danke hierfür nochmals.

Ein weiteres Dankeschön möchte ich an David Fritsche und Bianca Liebscher für Ihre Hilfestellungen bei Programmierfehlern bzw. –aufgaben richten.

Ein ganz besonderes Dankeschön geht an meine Familie, auf die ich zu jeder Zeit zählen kann und die mich während des gesamten Bachelors unterstützte.

1. Einleitung

Die Bachelorarbeit entstand im Rahmen einer Kooperation mit der in Dresden ansässigen Firma Biotype Diagnostic GmbH.

Die 1999 gegründete Firma gilt als Spezialist für molekulare Diagnostik und ist in der Forschung & Entwicklung, aber auch der Produktion und dem Vertrieb tätig. Verstärkt ist die Biotype Diagnostic GmbH auf dem Gebiet der Human-, Veterinär- und Pilzdiagnostik aktiv. [Dresdener Transferbrief] Sowohl dermatologische, als auch hämatologische/onkologische Fragestellungen können durch die Testsysteme ihres Produktportfolios beantwortet werden, wobei die Systemlösungen von der Probenentnahme bis zur Auswertung reichen. [URL 1] [Dresdener Transferbrief] Grundlage der meisten Testsysteme bildet die Multiplex-PCR, bei der simultan mehrere Reaktionen zur Vervielfältigung der DNA stattfinden, es folgt eine Detektion durch Kapillargelelektrophorese, Agarosegelelektrophorese oder mittels eines Microarrays [URL 1].

Die Sepsis stellt eine der größten Herausforderungen für einen Intensivmediziner dar. Sie ist definiert als „die Gesamtheit der lebensbedrohlichen klinischen Krankheitserscheinungen und pathophysiologischen Veränderungen als Reaktion auf die Aktion pathogener Keime und ihrer Produkte [...]“. [Werdan 2005] In Deutschland erkranken pro Jahr 154.000 Einwohner an septischen Erkrankungen und mit 60.000 Todesfällen stellen sie die dritthäufigste Todesursache dar. [Bruhn 2008]

Bei 30 % kann bislang auch noch kein mikrobiologischer Nachweis erfolgen [Deutschen Sepsis-Gesellschaft e. V. 2010] In der folgenden Arbeit liegt die Spezialisierung auf der Artspezifizierung von *Candida spp.* aus dem Mund- und Urogenitalbereich. Als *Candida*-Arten, die eine Sepsis verursachen, müssen folgende identifiziert werden: *Canida albicans*, *Candida glabrata*, *Candida guilliermondii*, *Candida krusei*, *Candida parapsilosis*, *Candida tropicalis*. [Plettenberg 2010] [Miksits 2004]

Bislang existieren nur zwei etablierte Methoden zur artspezifischen Pilzidentifizierung: Zum einen kann dies über die morphologische Untersuchung von Pilzkulturen geschehen und zum anderen ist eine Sequenzierung spezifischer Sequenzen und ein anschließender Vergleich mit der Datenbank möglich. Doch diese Verfahren sind entweder sehr langwierig (Kultivierung von Pilzen dauert oft Wochen), benötigen umfangreiches Fachwissen, können nicht immer die genaue Art des Pilzes bestimmen oder sind störanfällig bei Verunreinigungen bzw. sind erst gar nicht durchführbar (eine Sequenzierung kann fehlschlagen oder qualitativ schlechte Rohdaten erzeugen und eine Kultivierung ist auch nicht immer möglich). Durch den Einsatz eines Microarrays ist jedoch eine schnelle, einfache und speziesge-

naue Identifizierung möglich. Zudem können Mischproben analysiert werden, ohne eine vorherige Isolierung der einzelnen Pilzarten vornehmen zu müssen.

Ziel war es, mit Hilfe von spezifischen Pilz-Oligonucleotid-Sequenzen ein Diagnostik-Kit in Form eines Microarrays zu entwickeln, wobei besonderer Wert darauf gelegt werden sollte, dass es zu keinen falsch-positiven Ergebnissen mit dem Menschen (*Homo sapiens*) und einigen anderen Pilzen kommt, die auch bei gesunden Menschen in diesen Regionen vorkommen (*Saccharomyces cerevisiae*, *Penicillium spp.*, *Aspergillus spp.*, *Cladosporium spp.*, *Aureobasidium spp.*, *Alternaria spp.*, *Fusarium spp.*).

2. Grundlagen

2.1 DNA-der Träger unserer Vergangenheit

2.1.1 Allgemeines

Die Desoxyribonukleinsäure (internat./engl. DNA, desoxyribonuclein acid) ist die Trägerin der gesamten Erbinformationen. Sie wurde erstmals im Jahr 1869 von dem Baseler Arzt Friedrich Miescher entdeckt, der bei der Behandlung von Leukocyten aus dem Eiter gebrauchter Verbandsmaterialien eine Substanz bemerkte, die nicht von Proteasen abgebaut werden konnte. [Horton 2008] [Renneberg 2007] Ohne es zu wissen entdeckte er dabei den Träger der genetischen Information [Horton 2008]

Die Identifizierung dieser Funktion gelang 1944 der Forschungsgruppe um Frederick Griffith und Oswald Avery indem sie die DNA eines toxischen Bakteriums mit der eines nichttoxischen Stammes vermischten. Der ursprünglich harmlose Stamm übernahm dadurch die toxischen Eigenschaften. Diese Experimente bildeten neben dem Funktionsnachweis der DNA auch die Geburtsstunde der Gentechnik. [Renneberg 2007] [Horton 2008 a.])

Im Jahr 1953 veröffentlichten James Watson und Francis Crick das Molekülmodell der DNA (siehe Abbildung 2-1), welches zum Symbol der Molekularbiologie (Campbell 1998) und dem Dreh- und Angelpunkt der sogenannten Bio-Revolution wurde. [Renneberg 2010]



Abbildung 2-1: historische Bilder der DNA

links: Abbildung der DNA aus dem Originalpaper von 1953; rechts: Watson und Crick beim Betrachten eines DNA-Modells

2.1.2 Aufbau

Die DNA gehört zur chemischen Gruppe der Nukleinsäuren und repräsentiert damit eine Hauptklasse der biologischen Makromoleküle. Nukleinsäuren sind Poly-

mere aus Nukleotiden [Horton 2008 a.]). Diese wiederum bestehen aus einer Base, einer Pentose¹ und einem Phosphatrest. [Renneberg 2007] [Clark 2009] Im Fall der DNA stellt die Desoxyribose den Zucker dar. [Renneberg 2007] Derivate der Purine bzw. der Pyrimidine bilden die Basen der Nukleinsäuren. [Horton 2008 a.)] Die Basen, welche kovalent² an die Desoxyribose binden, sind die Pyrimidinbasen Cytosin (C) bzw. Thymin (T) und die Purinbasen Adenin (A) bzw. Guanin (G). Die Phosphatgruppe bildet ebenfalls eine kovalente Bindung zum Zucker. Sind Zucker und Base aneinander gebunden, liegt ein Nukleosid vor, nach der Bindung von Phosphat, Nukleotid. [Regenass-Klotz 2005]

Die Phosphatgruppe verknüpft, unter Ausbildung einer Phosphodiesterbindung zwischen der OH-Gruppe am sogenannten 3'-C-Atom des einen und der OH-Gruppe am 5'-C-Atom des anderen, zwei Zuckermoleküle und bildet damit ein Polymer aus Nukleotiden. Das resultierende Oligonukleotid (siehe Abbildung 2-2)

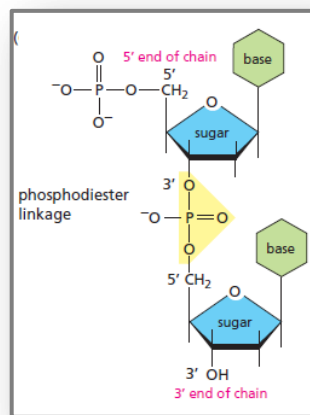


Abbildung 2-2: Oligonucleotids aus zwei Nukleotiden [Zvelibil 2008]
 Blau dargestellt ist der Zucker, grün die Base, gelb die Phosphatgruppe, welche die Phosphodiesterbindung bildet und rot das 5'-Ende bzw. 3'-Ende

besitzt ein Ende mit einem freien Phosphatrest, welches als 5'-Ende bezeichnet wird und ein Ende mit einer freien Hydroxylgruppe, welches als 3'-Ende bezeichnet wird. [Zvelebil 2008] Die Nukleotidsequenz ist als die Abfolge ihrer Nukleotide vom 5'- zum 3'-Ende definiert. Hierzu wird ein Einbuchstabencode genutzt, der nur die Basen angibt, beginnend am 5'-Ende, z. B. 5' -ACACCGTATG- 3'. [Zvelebil 2008] Diese Oligonukleotidkette wird als Rückgrat oder auch Backbone der DNA bezeichnet. [Regenass-Klotz 2005]

In der Natur kommt die DNA üblicherweise als Doppelstrang, die Doppelhelix, vor. [Regenass-Klotz 2005] Hierbei sind alle Basen auf der Innenseite dieses Stranges lokalisiert und die durch Phosphat verbundenen Zuckermoleküle befinden sich auf der Außenseite und bilden das Rückgrat (internat./engl. backbone) der DNA. Die

¹ Zucker, der aus fünf Kohlenstoffatomen gebildet wird

² vorherrschende Bindungsart in Nichtmetallverbindungen sowie Komplexen. Die Bindungsenergie liegt bei 200-400 kJ/mol

Basen der einen Oligonukleotidkette binden sich hierbei, entsprechend der sogenannten Watson-Crick-Basenpaarung, eine spezifische Purinbase mit einer spezifischen Pyrimidinbase mit denen des zweiten DNA-Stranges. Entsprechend dieser Regel bindet Adenin nur mit Thymin und Guanin nur mit Cytosin. Demnach besitzt jeder DNA-Strang eine dem Partnerstrang komplementäre Basenabfolge. Die Interaktion der einzelnen Basen erfolgt mittels Wasserstoffbrückenbindungen, wobei zwischen Adenin und Thymin zwei und zwischen Guanin und Cytosin drei Bindungen ausgebildet werden (siehe Abbildung 2-3).

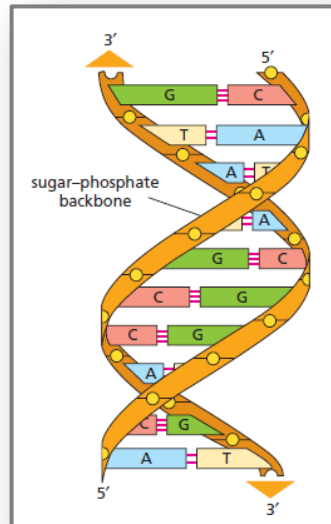


Abbildung 2-3: Darstellung der DNA [Zvelibil 2008]

Orange dargestellt ist das Zucker-Phosphat-Backbone, grün die Base Guanin, rot die Base Cytosin, gelb die Base Thymin, blau die Base Adenin; die Basen sind über drei bzw. zwei Wasserstoffbrückenbindungen verknüpft (rote Striche)

Wasserstoffbrückenbindungen zählen zu den nichtkovalenten Bindungen und sind somit in Biomolekülen schwächer als die kovalenten. Dadurch ist es leicht, die DNA-Helix in ihre Einzelstränge zu separieren, wie es z. B. bei der Polymerase-Kettenreaktion geschieht.

1.2 Polymerase-Kettenreaktion

1.2.1 Allgemeines

Im Jahr 1983 entwickelte Kary Mullis bei dem Biotechnologieunternehmen Cetus die Polymerase-Kettenreaktion (internat./engl. PCR, polymerase chain reaction) [Clark 2009], ein Verfahren zur in vitro³-Vervielfältigung von Desoxyribonukleinsäure und revolutionierte damit die Molekularbiologie. [Clark 2009] Da sie einen enormen Einfluss auf die Biologie und die Wissenschaft im Allgemeinen hatte,

³ lat.: „im Glas“, außerhalb des Organismus‘

wurde sie bereits 1993 mit dem Nobelpreis für Chemie gewürdigt. [Schmuck 2008] [Clark 2009] Ihre Anwendungsmöglichkeiten sind vielfältig. So ist es beispielsweise möglich, sie in der Forensik zur Identifikation von Opfern bzw. Tätern zu nutzen. Infektiöse Krankheiten, wie z. B. AIDS, können noch vor dem Eintreten der Symptomatik nachgewiesen werden. [Clark 2009] Durch kleinste DNA-Mengen von z. B. ausgestorbenen Arten können Verwandtschaftsgrade geklärt und Hinweise auf die Stammesentwicklung gewonnen werden. Man könnte mit Hilfe der PCR sogar längst ausgestorbene Arten wiedererschaffen, sofern genügend DNA vorhanden ist. [Renneberg 2010]

Die Polymerasekettenreaktion findet seit ihrer Entwicklung in allen Bereichen der Biowissenschaften Anwendung. [Clark 2009]

1.2.2 Ablauf

Der Ablauf, welcher mittlerweile in einem Thermocycler⁴ stattfindet, besteht im Wesentlichen aus drei Schritten: der *Denaturierung*, dem *Annealingschritt* und der *Elongation*. [Mühlhardt 2009]

Denaturierung

Unter thermischer Einwirkung (92-95 °C) erfolgt für 0,5-2 min die Trennung der doppelsträngigen DNA in ihre Einzelstränge. [Linz 1990] [Pestana 2010] Bei GC-reichen DNA-Strängen sollte die Dauer auf 3-4 min erhöht werden. [Pestana 2010]

Annealing/Hybridisierung

Anschließend hybridisieren bei etwas niedrigerer Temperatur die zwei Oligonukleotide an der Template-DNA und fungieren an ihrer 3' Flanke als Primer für die DNA-Polymerase. [Linz 1990] Dieser Schritt sollte 0,5-2 min dauern und die optimale Annealing Temperatur ergibt sich laut [Rychlick 1990] aus Formel (1). [Pestana 2010] [Mühlhardt 2009]

$$T_a^{OPT} = 0,3T_m^{Primer} + 0,7T_m^{Produkt} - 14,9 \quad (1)$$

T_a^{OPT}	...optimale Annealingtemperatur
T_m^{Primer}	...Schmelztemperatur des Primers mit der niedrigsten Schmelztemperatur
$T_m^{Produkt}$...Schmelztemperatur des PCR-Produktes

Elongation

Nun folgt bei dem Temperaturoptimum der DNA-Polymerase (Taq-Polymerase 72 °C) dessen Anlagerung an die Primer und mittels der dNTPs vom 3'-Ende des Primers ausgehend die Primer-Verlängerung (internat./engl. primer extension)/die Zweitstrangsynthese.[Hermey 2010]

⁴ Programmierbarer Heizblock, der einen beheizten Deckel besitzt und auf schnelle Temperaturänderungen ausgerichtet ist [Hermey..]

Dieser Zyklus (Denaturierung-Annealing-Elongation) findet nun mehrfach, meist 20-30 Mal statt. [Linz 1990] [Hermey 2010] Zur Verdeutlichung des Ablaufes siehe Abbildung 2-4.

Der initiale, als auch terminale Schritt der Polymerase-Kettenreaktion verläuft in etwas modifizierter Weise zum normalen Zyklusverlauf. Bei der initialen Denaturierung muss gewährleistet werden, dass die Template-DNA vollständig denaturiert wird. Dies geschieht durch eine verlängerte Denaturierungsphase. [Hermey 2010] Ein nicht vollständig denaturierter DNA-Strang würde zu einer ineffizienten Elongation im ersten Cycle führen, was wiederum in einer sehr schlechten PCR-Produkt-Ausbeute resultieren würde. Bei GC-Bereichen $\leq 50\%$ sollte die initiale Denaturierung 1-3 min dauern. Bei GC-reicheren Sequenzabschnitten bis zu 10 min.

Ebenso sollte die finale Elongation etwas länger dauern (5-15 min), um sicher zu stellen, dass alle DNA-Produkte vollständig verlängert wurden. [Hermey 2010] [Pestana 2010]

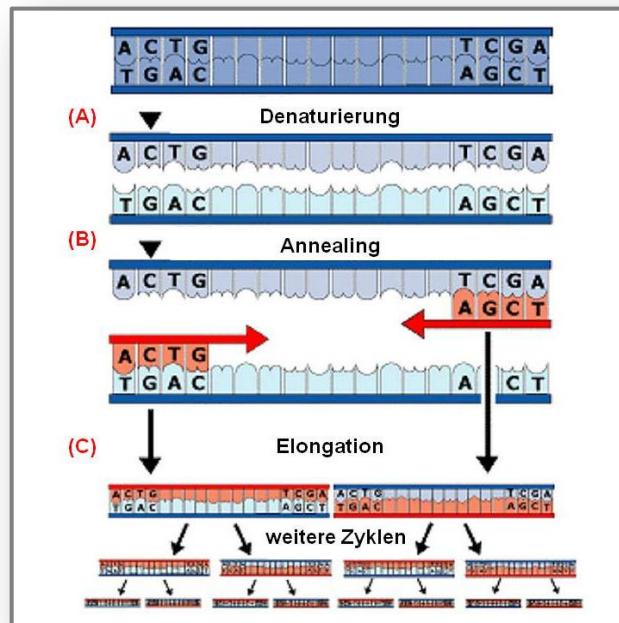


Abbildung 2-4: Schritte der PCR [URL 2]

Von oben beginnend werden die Schritte der PCR verdeutlicht: (A) die Denaturierung, (B) die Annealing-Phase und (C) die Elongation; anschließend folgen weitere dieser Zyklen

Der für die Durchführung benötigte Reaktionsansatz setzt sich aus einer temperaturstabilen DNA-Polymerase (EC 2.7.7.7), meist *Taq*-DNA-Polymerase, der zu amplifizierenden [=vervielfältigen] doppelsträngigen DNA (internat./engl.Template-DNA), zwei kurzen zur DNA komplementären Oligonukleotiden im Überschuss, einer Mischung aus Desoxynucleosidtriphosphaten (dNTPs) und Magnesiumio-

nen, welche als Kofaktoren für die DNA-Polymerase fungiert, zusammen [Clark 2009]

Nach dem Ablauf der Polymerase-Kettenreaktion kann das Reaktionsprodukt gelelektrophoretisch über die Größen der Fragmente analysiert werden oder es kann eine Analyse mittels Hybridisierung mit spezifischen Sonden auf z. B. einem Microarray erfolgen. [Linz 1990]

1.3 Microarray

1.3.1 Allgemeines

Mit der Idee, tausend DNA-Sonden auf einem einzelnen Glas-Mikrochip aufzubringen, entstanden in den frühen 90er Jahren die ersten Microarrays durch die Firma Affymetrix. [Renneberg 2010] Die Ziele, welche die Entwicklung der Microarrays verfolgten, waren der Hochdurchsatz mit dem damit verbundenem Zeit- und Kostenersparnis.

Allgemein versteht man unter dem Begriff Microarray eine Anordnung von Molekülen, die auf einer Fläche, welche das Mikrotiterplattenformat (MTP)⁵ unterschreitet, deponiert sind.

Mittlerweile ist es sogar möglich, ein Microarray herzustellen, welches auf der Fläche von einem cm² mehr als 200.000 Spots (separierte „Punkte“) besitzt. [Müller 2004]

Die Anwendungsmöglichkeiten der Microarrays sind enorm. Beispielhaft zu nennen sind hierbei Mutationstests eines DNA-Abschnittes, Identifikationen von Unterschieden auf genomischer⁶ Ebene, Quantifikationen von Protein-Proteininteraktionen und SNP-Analyse⁷. [Müller 2004] [Renneberg 2010] Klassifizieren kann man Microarrays entweder nach ihrer Anwendung oder nach der Art der exponierten Probe. [Müller 2004]

1.3.2 DNA-Microarray

Das grundlegende Prinzip der DNA-Microarrays basiert darauf, dass zwei Nukleinsäuren hybridisieren, wenn sie komplementär zueinander sind. Dieses Prinzip kann zur quantitativen Erfassung einer unbekannten Nukleinsäure genutzt werden, indem die Menge der komplementären Sequenz, die hybridisiert hat, gemessen wird. Diese Messung erfolgt mittels der Detektion eines chemischen Markers, welcher an die unbekannte Nukleinsäure, das Target oder an die komplementären Oligonukleotide, die Sonden, gebunden ist. Die Sonden sind hierbei auf der Oberfläche des Microarrays immobilisiert und hybridisieren mit einer ganz bestimmten Sequenz. [de Rinaldis 2006] Die DNA-Microarrays lassen sich in drei Gruppen, entsprechend der Sondelänge, einteilen. So kann man ganze PCR-Produkte auf

⁵ Meist 100-200 µl Reaktionsvolumen [Antranikian 2006]

⁶ Genom = Gesamtheit aller Gene/Erbinformationen

⁷ SNP = Mutation oder Veränderung eines einzelnen Nucleotids

einem Microarray immobilisieren, sprich Nukleinsäuren von 100-1000 bp Länge, aber auch Oligonukleotide. Letztere lassen sich wiederum in kürzere (20-25 bp) und längere (>50 bp) Oligonukleotide einteilen. Microarrays mit PCR-Produkten werden meist bei Organismen genutzt, die noch nicht sequenziert wurden, wodurch auch keine Sonden im Vorfeld generiert werden können. Bei Expressionsstudien werden vorwiegend Sonden mit längeren Oligonukleotiden verwendet. Die kurzen Oligonukleotid-Sonden kommen z. B. bei SNP-Analysen zur Anwendung. [Müller 2004]

1.4 Strategie zur Identifikation der Pilze

Microarrays repräsentieren eine sensitive und schnelle Methode zur Identifikation von Pilzen. Eine Möglichkeit dies zu realisieren ist es, variable Bereiche eines Pilz-Genoms mittels PCR zu amplifizieren und anschließend in denaturierter, einzelsträngiger Form auf dem Microarray, wo sie als Sonden fungieren, zu immobilisieren. [Busch 2010] Eine weitere Möglichkeit ist, Sonden bereits im Vorfeld auf dem Microarray zu immobilisieren und die amplifizierte Pilz-DNA damit hybridisieren zu lassen. [Strandhagen 2011] [Busch 2010]

Als mögliche Zielsequenzbereiche wurden β -Tubulin-DNA, Chitinsynthase I-DNA und das rRNA-Operon gewählt.

β -Tubulin bildet mit α -Tubulin den Hauptbestandteil der Tubuline. Diese wiederum sind die Grundbausteine der Mikrotubuli, welche die für die Verteilung der Zellorganelle und Makromoleküle im Zellinneren, sowie die Polarität der Bewegung, als auch für die Prägung der Zellform und deren Erhaltung verantwortlich sind. [Wolf 1997] [Hirsch-Kaufmann 2009] Die Gene, welche für diese Proteine codieren, sind hochkonserviert, jedoch kann es in Folge einer Anpassung auch zu Sequenzabweichungen kommen. [Gabrio 2001] [Wolf 1997] Da Mikrotubuli obligate eukaryontische Zellbestandteile sind, können bei der Verwendung von β -Tubulinen als Zielsequenz, Verunreinigungen durch Prokaryonten, wie z. B. Bakterien, zu keinem falsch-positiven Ergebnis führen.

Chitinsynthase I zählt zu den Chitinsynthasen, den Enzymen, welche die Elongation der Chitinketten katalysieren. [Griffin 1994] Die Enzyme bestehen aus einem N-terminalen, hydrophilen, nicht konservierten Bereich, einem neutralen, hochkonservierten mittlerem Abschnitt und einem hydrophoben, konservierten C-Terminus. [Brakhage 1990] Chitin und damit Chitinsynthase kommt nur in Atropoden⁸, die es zum Bau ihres Exoskelettes nutzen und in Pilzen für Bildung der pilzlichen Zellwand vor. [Buselmaier 2006] Durch diesen Fakt könnte man bei der Verwendung von Chitinsynthase-Targets Kreuzreaktionen auf dem Microarrays mit Prokaryonten oder dem Menschen gleich zu Beginn ausschließen. Teile der **rDNA**, bei Prokaryonten die 16S-, 5S-, 23S und bei Eukaryonten die 18S-, 5,8S- und die 28S-rDNA, codieren für ribosomale DNAs, welche 85 % der zellulären RNA und 40 % des Gewichtes eines Ribosomen darstellt. [Gabrio 2001]

⁸ Insekten, Spinnentiere, Krebse, Tausendfüßler

[Griffin 1994] Zwischen diesen hochkonservierten Bereichen befinden sich DNA-Abschnitte mit hohem GC-Gehalt, deren Aufgabe es ist, die einzelnen rDNA-Einheiten zu separieren. [Graw 1995] Die für uns wichtigen Spacerregionen werden als Internal transcribed spacer I (ITSI) und als ITSII bezeichnet und befinden sich zwischen der 18S- und 5,8S- und zwischen 5,8S- und 28S-rDNA. [Tropp 2008] Sie besitzen eine niedrigere Konservierung als die rDNA-Gene. [Werner 2002] Besonders geeignet ist die rDNA, da sie, aufgrund der benötigten Menge, sehr oft vorkommt und somit analytisch leicht zu erfassen ist. [Buselmaier 2006] Die Variabilität bei den Spacerregionen eignet sich zur Differenzierung nahe verwandter Arten und die rDNA-Gene eignen sich aufgrund ihrer hohen Konservierung zu Differenzierung von Organismengruppen. [Arora 2009] [Caspary 2006]

Voraussetzung für die Nutzung von Micorarrays sind Kenntnisse über die Sequenzbereiche. [Busch 2010] Da sich herausstellte, dass die Datenlage bei β -Tubulin und Chitinsynthase I sehr schlecht ist, erfolgte im Weiteren eine Fokussierung auf rDNA, speziell 18S-ITS I-5,8S-ITS II-28S. Zur Sequenzsuche wurden die biologischen Datenbanken des National Center for Biotechnology Information (NCBI) und SILVA genutzt.

Das **NCBI** wurde 1899 als ein Institut zur Speicherung und Verarbeitung molekularbiologischer Daten gegründet. Die Daten des NCBI stammen aus anderen Datenbanken. Genbank, die genutzte Datenbank, ist eine öffentliche Sequenzdatenbank, die sowohl Protein-, als auch Nukleotidsequenzen beinhaltet. [Gaedeke 2007]

Die **SILVA** –Datenbank bietet qualitativ überprüfte und alignierte Datensätze für rRNA-Sequenzen. [Pruesse 2007]

Da 18S-ITS I-5,8S-ITS II-28S nur in Teilsequenzen vorzufinden waren, mussten diese entsprechend zusammengefügt, assembliert, werden. Hierzu wurde das Programm *CLC Main Workbench* des Unternehmens CLC bio A/S⁹ genutzt. Die Datenbankrecherche, als auch das Assemblieren der gefundenen Sequenzen wurde von Doreen Kropp, einer Kommilitonin, im Rahmen ihrer Bachelorarbeit vorgenommen.

Anschließend konnte auf Basis dieser Daten mit der Generierung der Primer und anschließend der Sonden begonnen werden.

⁹ <http://www.clcbio.com>)

3. Oligonukleotide

Damit ein Oligonukleotid als Primer oder Sonde geeignet ist, müssen bei deren Generierung verschiedene Aspekte beachtet werden.

Die Primer, als auch Sonden sollten grundsätzlich **spezifisch** sein, um in unserem Fall eindeutig ganz bestimmte Pilzarten zu identifizieren. [Dieffenbach, 2003] Es sollte hierbei gewährleistet sein, dass es zu keinen fälschlicherweise positiven Ergebnissen mit dem Menschen (*Homo sapiens*), *Saccharomyces cerevisiae*, *Penicillium spp.*, *Aspergillus spp.*, *Cladosporium spp.*, *Aureobasidium spp.*, *Alternaria spp.* oder *Fusarium spp.* kommt.

Eine **Länge** von mindestens 18 Nukleotiden gewährleistet eine spezifische Hybridisierung. [van Pelt-Verkuil 2008] Dies ergibt sich aus der mittleren Wahrscheinlichkeit für das Auftreten einer Base. Empirisch fand man den Wert, 0,283. Um für das menschliche Genom (ca. 3 Milliarden Basenpaare) eine spezifische Sequenz zu erzeugen, benötigt man demnach 18 Nukleotide: $0,283^{18} = 1/(7 \times 10^9)$ [Lin 1990] Jedoch sollten Primer eine Länge von 30 Nukleotiden nicht überschreiten, da mit steigender Länge auch die Wahrscheinlichkeit an Fehlhybridisierungen innerhalb des Primers zunimmt. [van Pelt-Verkuil 2008] [Mülhardt 2009] Mit zunehmender Nukleotidzahl kommt es zu einer steigenden Schmelztemperatur und dies führt zu einer höheren Annealingtemperatur (siehe Formel 1). Trotz dass eine thermostabile DNA-Polymerase verwendet wird, verliert diese bei jeder Erhitzung an Aktivität. [Hermey 2010] Auch bei Sonden steigt mit zunehmender Länge die Spezifität, so besitzt z. B. eine 60 bp lange Sonde die achtfache Sensitivität einer 25 bp langen Sonde. [Chou 2004] Jedoch steigt auch mit zunehmender Länge die Möglichkeit, dass ein zufälliges DNA-Fragment eine passende Hybridisierungsstelle findet. [Kimmel 2006]

Der **Gehalt an Guanin/Cytosin** sollte etwa 50 % betragen, da diese Basen laut Watson-Crick-Paarung über drei Wasserstoffbrückenbindungen hybridisieren (siehe Abbildung 3-5) und somit eine stärkere Bindung des Primers oder der Sonde an die Template-DNA realisieren als Adenin/Thymin. [Dieffenbach 2003]

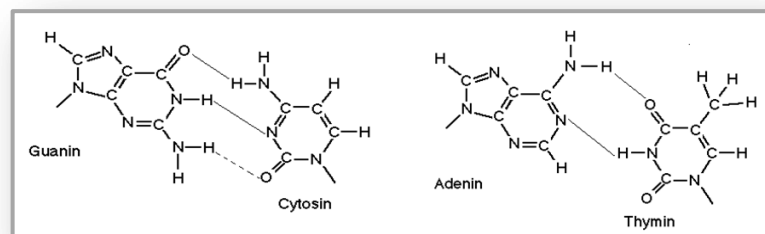


Abbildung 3-5: Watson-Crick-Basenpaarung [URL 3]

Bei Regionen mit niedrigem GC-Gehalt ist es daher empfehlenswert, Primer/Sonden zu generieren, die außerhalb dieser GC-schwachen Bereiche hybridisieren. Ein Gehalt von weit über 65 % ist jedoch auch nicht empfehlenswert, da dieser zur Formierung von internen Haarnadeln oder einer starken Bindung mit Nicht-Target-DNA führen kann. [van Pelt-Verkuil 2008] [Kimmel 2006] Desweiteren haben DNA-abhängige Polymerasen große Schwierigkeiten, Regionen mit einem hohen GC-Gehalt zu amplifizieren. [van Pelt-Verkuil 2008] [Dorak 2007]

Homopolymere Sequenzen, wie z. B. –AAAA–, sind zu vermeiden, da diese die Spezifität der Primer bzw. Sonden senken und somit auch zu Fehlhybridisierungen führen können. Sekundärstrukturen sollten generell vermieden werden, da diese die Hybridisierungseffizienz herabsetzen und zu falschen Amplifikationen (siehe Abbildung 3-6), den sogenannten Dimeren, führen können. [Mülhardt 2009]

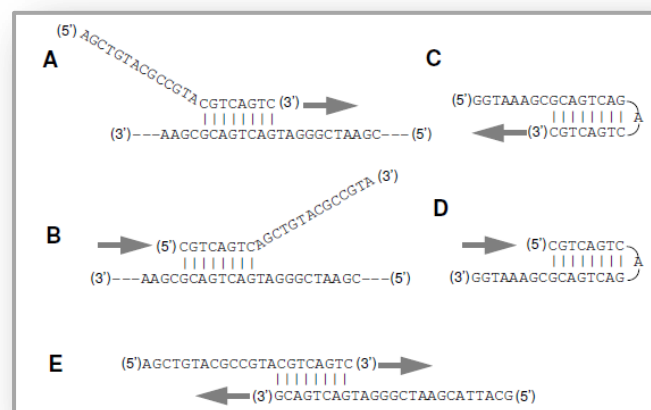


Abbildung 3-6: mögliche Fehlhybridisierungen bei Primern

A-E zeigen mögliche Fehlhybridisierungen bei Primern. Die grauen Pfeile zeigen die Richtung, in der die Verlängerung stattfindet [Mülhardt 2009]

Eine weitere Möglichkeit um Dimere zu verhindern, ist der Einsatz von Guanin- oder Cytosin-Analoga, deren Bindungen wesentlich geringer sind als bei Guanin-Cytosin-Hybridisierung (siehe hierzu [Lahoud 2008]). Hierdurch kann man auch Primer/Sonden in GC-reichen Sequenzbereichen generieren. [van Pelt-Verkuil 2008]

Da am **3'-Ende des Primers** die DNA-Polymerase bindet und somit die Elongation beginnt ist es ratsam, diese Stelle möglichst stabil zu gestalten. Dies kann z. B. dadurch erreicht werden, dass sich am 3'-Ende ein oder zwei Guanin/Cytosin befinden. Eine sehr hohe Anzahl an GC ist nicht empfehlenswert, da dies wieder zu Fehlhybridisierungen führen könnte. [Mülhardt 2009]

Eine sehr wichtige Komponente ist die **Hybridisierungseffizienz**. Sie sollte bei beiden Primern gleich sein, um eine asymmetrische Amplifikation, d. h. die Amplifikation nur eines Stranges, zu vermeiden. Ebenso sollte sie bei allen auf den Microarray befindlichen Sonden ähnlich sein, um die Hybridisierung mit allen Sonden

gleichermaßen zu ermöglichen. Sie ist von der Schmelztemperatur und der Neigung zur Bildung von Sekundärstrukturen abhängig. [Strandhagen, 2010]

Die **Schmelztemperatur** ist die Temperatur, bei der die Hälfte der Primer/Sonden an die Target-DNA gebunden sind. Die Neigung Sekundärstrukturen auszubilden wird mit der *Gibbs'schen freien Energie (G)* zum Ausdruck gebracht. [Kennedy 2011] [Strandhagen 2010]

Die **Gibbs'sche freie Energie, G** beschreibt die Energie von Molekülen in Lösung. Die Änderung von G während einer Reaktion wird als *freie Reaktionsenthalpie, ΔG* bezeichnet (Formel (2)).

$$\Delta G = G_{\text{Produkt}} - G_{\text{Reaktanten}} \quad (2)$$

ΔG	...freie Enthalpie
G _{Produkt}	...absolute Gibbs'sche freie Energie der Produkte
G _{Reaktanten}	...absolute Gibbs'sche freie Energie der Reaktanten

Da die absolute Gibbs'sche freie Energie nicht messbar ist, wird zur Bestimmung von ΔG folgende Formel 3 genutzt.

$$\Delta G = \Delta H - T\Delta S \quad (3)$$

ΔG	...freie Enthalpie
ΔH	...Enthalpieänderung
T	...Temperatur [K]
ΔS	...Entropieänderung

Als **Enthalpieänderung ΔH** wird die Änderung des Wärmehaltes während einer Reaktion bezeichnet. Bei ΔH<0 wird während der Reaktion Wärme an die Umgebung abgegeben, bei ΔH>0 aufgenommen.

Die **Entropieänderung ΔS** stellt ein Maß für die Änderung der Ordnung eines Systems während einer Reaktion dar. Bei ΔS<0 nimmt die Ordnung des Systems zu, bei ΔS>0 nimmt sie ab. [Horton 2008]

Oftmals ist es sinnvoller, Primer mit ähnlichen Werten für ΔG zu generieren, als nur auf ähnliche Schmelztemperaturen zu achten, da es vorkommen kann, dass forward und reverse Primer unterschiedliche Bindungsenthalpien besitzen, was zu divergentem Verhalten nach dem Erreichen der Schmelztemperatur führt (siehe Abbildung 3.7). [Yuryev 2007] Das Verhalten bei der Annealingtemperatur, die wie in Kapitel 1.2.2 beschrieben, ausgehend von der Schmelztemperatur gebildet wird, wäre in diesem Fall nicht identisch mit dem der Schmelztemperatur. Hierdurch wird die Amplifikation eines Stranges begünstigt und verläuft effizienter. Diese einseitige Amplifikation wird mit jedem Zyklus wiederholt und verstärkt. Dies führt zu einem ineffizienten PCR-Ergebnis und kann in der Bildung von Artefakten resultieren. [Yuryev 2007]

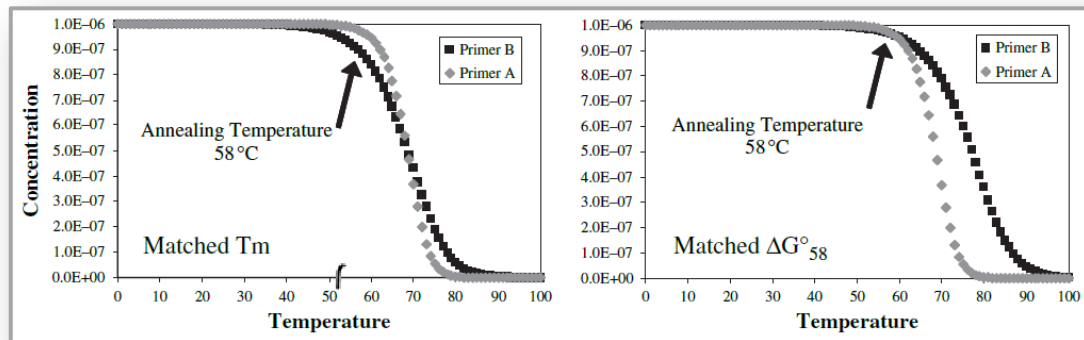


Abbildung 3-7: Darstellung von zwei unterschiedlichen Hybridisierungskurven während einer PCR [Yuryev 2007]

Hierbei sind Hybridisierungseffizienzen von zwei unterschiedlichen Primern in Abhängigkeit von der Temperatur dargestellt. Die Annealingtemperatur wurde links, nur anhand der Schmelztemperatur bestimmt und rechts erfolgte eine neue Generierung von Primer B, so dass beide identische Werte für ΔG_{58} aufweisen.

Zur Berechnung der Schmelztemperatur gibt es sehr viele Möglichkeiten, allerdings sind alles nur Annäherungen an den realen Wert.

Für Primer ist die empfohlene Berechnung die Nearest-Neighbor Methode nach Breslauer, 1986. [Dorak 2007] [Mühlhardt 2009] [Yuryev 2007]

$$T_m(\text{oligo}) = \frac{\Delta H(\text{oligo})}{\Delta S(\text{oligo}) + R \ln \left(\frac{c}{4} \right)} + T_0 \quad (4)$$

$\Delta H(\text{oligo})$...Enthalpieänderung [k*cal/mol]

$\Delta S(\text{oligo})$...Entropieänderung [cal/k*mol]

...allgemeine Gaskonstante

c ...Konzentration des Oligonucleotids [mol/l]

T_0 ...absoluter Nullpunkt, -273,15 °C

Hierbei werden durch ΔH und ΔS die thermodynamischen Parameter benachbarter Basen mit in Betracht gezogen, wie in Formel(5) und (6) zu sehen. [Dorak 2007] [Breslauer 1986]

$$\Delta H(\mathbf{p}) = \sum_i n_i \Delta H(i) + \Delta H(\text{init./term. GC}) + \Delta H(\text{init./term. AT}) + \Delta H(\text{sym}) \quad (5)$$

$$\Delta S(\mathbf{p}) = \sum_i n_i \Delta S(i) + \Delta S(\text{init./term. GC}) + \Delta S(\text{init./term. AT}) + \Delta S(\text{sym}) \quad (6)$$

Bei der Publikation zur Berechnung Schmelztemperaturberechnung veröffentlichte [Breslauer 1986] Werte für ΔH und ΔS , welche kurze Zeit später von exakteren abgelöst wurden. Aus diesen entstand 1998 ein „unified NN set“ von SantaLucia, 1998 (siehe Tabelle 3-1). [Yuryev 2007]

Tabelle 3-1: allgemeingültige SantaLucia-Werte [SantaLucia 1986]

Sequence	ΔH kcal/mol	ΔS kcal/mol
5'-AA-3'/3'-TT-5'	-7,9	-22,2
5'-AT-3'/3'-TA-5'	-7,2	-20,4
5'-TA-3'/3'-AT-5'	-7,2	-21,3
5'-CA-3'/3'-GT-5'	-8,5	-22,7
5'-GT-3'/3'-CA-5'	-8,4	-22,4
5'-CT-3'/3'-GA-5'	-7,8	-21
5'-GA-3'/3'-CT-5'	-8,2	-22,2
5'-CG-3'/3'-GC-5'	-10,6	-27,2
5'-GC-3'/3'-CG-5'	-9,8	-24,4
5'-GG-3'/3'-TT-5'	-8	-19,9
init./term. GC	0,1	-2,8
init./term. AT	2,3	4,1
Symmetry	0	-1,4

Da die Entropie stark von Salzen beeinflusst wird, muss eine zusätzliche Salz-Korrektur-Formel [Ahsen 2001] zur Berechnung der einzelnen Entropien genutzt werden. [Dorak 2007]

$$\Delta S(\text{polymer NN}, [\text{Na}^+]) = \Delta S(\text{unified NN}, 1 \text{ M NaCl}) + 0,368 \times (N-1) \times \ln[\text{Na}^+] \quad (7)$$

$\Delta S(\text{polymer NN}, [\text{Na}^+])$...Entropie des Polymers NN unter Einbezug der Größe $[\text{Na}^+]$
$\Delta S(\text{unified NN}, 1 \text{ M NaCl})$...Entropie des Polymers NN nach „unified NN set“ [k*cal/mol]
N	...Länge des Oligonukleotids [bp]
$[\text{Na}^+]$...gesamte Salzkonzentration der Lösung [mmol/l]

Zur Erfassung der Gesamtsatzkonzentration in Lösung, $[\text{Na}^+]$ wird die Formel nach [Ahsen 2001] verwendet.

$$[\text{Na}^+] = [\text{Monovalent cations}] + 120(\sqrt{[\text{Mg}^{2+}] - [\text{dNTPs}]} \quad (8)$$

$[\text{Na}^+]$...Gesamtsatzkonzentration der Lösung [mmol/l]
[Monovalent cations]	...Konzentration der monovalenten Kationen [mmol/l]
$[\text{Mg}^{2+}]$...Konzentration der Magnesiumionen [mmol/l]
[dNTPs]	...Konzentration der dNTPs [mmol/l]

Die Schmelztemperatur der Primer sollte nicht zu hoch sein, da, wie schon beschrieben, die DNA-Polymerase bei höheren Temperaturen an Aktivität verliert. [Ahmed 2007]

Für Oligonukleotide mit einer Länge von über 40 Nukleotiden verliert die Berechnung nach der Nearest-Neighbor Methode an Gültigkeit. [Yuryev 2007] [Meinkoth 1984] Meinkoth entwickelte hierfür folgende Formel:

$$\mathbf{Tm(oligo) = 81,5 + 16,6(\log_{10} M) + 0,41(\%GC) - 0,61(\%form) - 500/N} \quad (9)$$

M	...Konzentration monovalenter Kationen [nmol/l]
%GC	...GC-Gehalt des Oligonukleotids
%form	...Formamidgehaltes in Lösung [% (v/v)]
N	...Länge des Oligonukleotids [bp]

Die genannten Kriterien sind nur als allgemeine Hinweise zu verstehen und je nach gestellter Anforderung an die Primer/Sonden sind Modifikationen, abweichend der gängigen Regeln, möglich bzw. unabdingbar.

4. Oligonukleotidtools im Internet

Um ein Programm zu finden, welches zur Generierung der Primer und der Sonden genutzt werden kann, erfolgte zunächst eine Auflistung aller potentiell geeigneten Internettools (Tabelle 4-2).

Tabelle 4-2: Auflistung der potentiell geeigneten Internetprogramme

Tool	Internetpräsenz	Empfohlen
BiSearch	http://probes.pw.usda.gov/batchprimer3/index.html	[Yuryev]
DoPrimer	http://doprimer.interactiva.de/	[Saikia 2008] [Abd-Elsalam 2003]
EPrimer3	http://mobyale.pasteur.fr/cgi-bin/portal.py?#forms::eprimer3	[Saikia 2008]
Gene Fisher	http://bibiserv.techfak.uni-bielefeld.de/genefisher2/	[Saikia 2008] [Abd-Elsalam 2003]
Medusa	http://www.cgr.ki.se/cgr/MEDUSA/	[Saikia 2008] [Abd-Elsalam 2003]
NetPrimer	http://www.premierbiosoft.com/support/faq/faq.html	[Deb]
Primer Design	http://pga.mgh.harvard.edu/servlet/org.mgh.proteome.Primer	[Saikia 2008] [Abd-Elsalam 2003]
Primaclade	http://www.umsl.edu/services/kellogg/primaclade.html	[Yuryev]
Primer3	http://frodo.wi.mit.edu/primer3/	[Saikia 2008] [Abd-Elsalam 2003] [Deb] [Yuuryev]
Primer3Plus	http://www.bioinformatics.nl/cgi-bin/primer3plus/primer3plus.cgi	[Hermey] S.33
Primer-BLAST	http://www.ncbi.nlm.nih.gov/tools/primer-blast/index.cgi?LINK_LOC=NcbiHomeAd	[Kennedy]
Primer-Quest	http://www.idtdna.com/biotools/primer_quest/primer_quest.asp	[Saikia 2008] [Abd-Elsalam 2003]
PrimerSelection	http://alces.med.umn.edu/rawprimer.html	[Saikia 2008] [Abd-Elsalam 2003] [Deb]
PrimoPro	http://www.changbioscience.com/primo/primo.html	[Saikia 2008] [Abd-Elsalam 2003] [Yuryev]
Raw Primer	http://alces.med.umn.edu/rawprimer.html	[Saikia 2008] [Abd-Elsalam 2003]
Primer Project	http://www-nmr.cabm.rutgers.edu/bioinformatics/Primer_Primer	[Saikia 2008] [Abd-Elsalam 2003]
Web Primer	http://genome-www2.stanford.edu/cgi-bin/SGD/web-primer	[Saikia 2008] [Abd-Elsalam 2003]

Anschließend wurden Kriterien erarbeitet, die ein für die gestellte Aufgabe geeignetes Programm besitzen muss und anhand dieser Merkmale wurde jedes Programm, unter Zuhilfenahme der jeweiligen Internetpräsenzen, einzeln evaluiert.

Existent: Um mit einem Programm zu arbeiten, muss es existieren. Einige der aufgelisteten Programme schienen jedoch verschwunden zu sein.

Ausschluss von: DoPrimer, Medusa, PrimerSelection, Primer Design, Raw Primer und WebPrimer.

Eigener Algorithmus: Weiterhin gab es auch einige Programme, deren Verfahren zur Ermittlung des besten Primerpaares vollständig von anderen Programmen übernommen wurde und lediglich Zusatzfunktionen beinhalteten. Wenn die zusätzlichen Funktionen mittlerweile in das verfahrensgebende Programm implementiert wurden, kann man die „Kopieprogramme“ ausschließen.

Ausschluss von: Primaclade, Primer-BLAST, Primer3Plus und PrimerQuest.

Berechnung der Schmelztemperatur mit den SantaLucia-Werten: Wie in den Grundlagen geschildert, liefert die Berechnung der Schmelztemperatur mit den SantaLucia-Werten die wahrheitsgetreuesten Werte, jedoch existieren Programme, die immer noch mit ursprünglichen Breslauer Werten rechnen.

Ausschluss von: BiSearch [Tusnady 2005] und NetPrimer .

Berechnung längerer Oligonukleotide möglich: Da das Ziel des Projektes ein Microarray sein soll und dafür Sonden benötigt werden, die auch über die gewöhnliche Primerlänge hinausgehen, muss das Programm auch mit längeren Oligonukleotiden rechnen können. Jedoch verliert die Nearest-Neighbor Methode ab 40 Nukleotiden ihre Gültigkeit [Yuryev 2007] und keines der genannten Programme nimmt eine Unterscheidung nach Nukleotidanzahl vor, wodurch letztendlich keines der aufgelisteten Programme den gestellten Anforderungen gerecht wurde.

Spezifitätsüberprüfung: Wie schon beschrieben, bestand die besondere Anforderung an die Primer bzw. Sonden, dass sie keinerlei Hybridisierung mit bestimmten anderen Organismen zulassen. Eine Möglichkeit hierzu bietet lediglich das Programm Primer-BLAST, das mittels des Tools BLAST (siehe hierzu Kapitel 5.3.8) in einer Sequenzdatenbank nach Übereinstimmungen mit dem Primerpaar sucht. Primer-BLAST basiert auf dem Algorithmus von Primer3 und mittlerweile wurde die BLAST-Funktion auch auf Primer3 übernommen. Jedoch bieten beide Programme keine direkte Auswertung der BLAST-Ergebnisse, wodurch sie für die spezielle Spezifitätsüberprüfung ungeeignet sind.

Verdeutlicht und zusammengefasst werden diese Ausschlüsse der einzelnen Programme in Tabelle 4-3.

Tabelle 4-3: Evaluierung der Internetprogramme

	Existent	Algorithmus erkennbar	eigener Algorith- mus	Verwendung SantaLucia- Werte	Berechnung längerer Oligos	Spezifi- täts- check
BiSerach	x	x	x			
DoPrimer						
EPrimer3	x	x				
Gene Fisher	x	x	x	x		
Medusa						
NetPrimer	x	x	x			
Prim'er Project	x	x	x			
Primaclade	x	x				
Primer Design						
Primer3	x	x	x	x		/
Primer3Plus	x	x				
Primer-Blast	x	x				/
PrimerDesign	x					
Primer-Quest	x	x				
Primer Selection						
PrimoPro	x	x	x	x		
Raw Primer						
WebPrimers	x					

Da sich keines der evaluierten Programme zur Lösung der gestellten Aufgabe eignete, musste ein eigener Algorithmus entwickelt werden.

5. AP²D

AP²D steht für „Algorithm for Primer-Probe Design“ und stellt einen selbstentworfenen Algorithmus dar, welcher die zuvor gestellten Anforderungen erfüllt und somit speziell auf die Generierung von Oligonukleotiden zur *Candida spp.*-Identifikation geeignet ist.

Der im eigenen Primerprogramm verwendete Algorithmus entstand in Anlehnung an [Kämpke 2001], [Rychlik 1989] bzw. [Montera 2008].

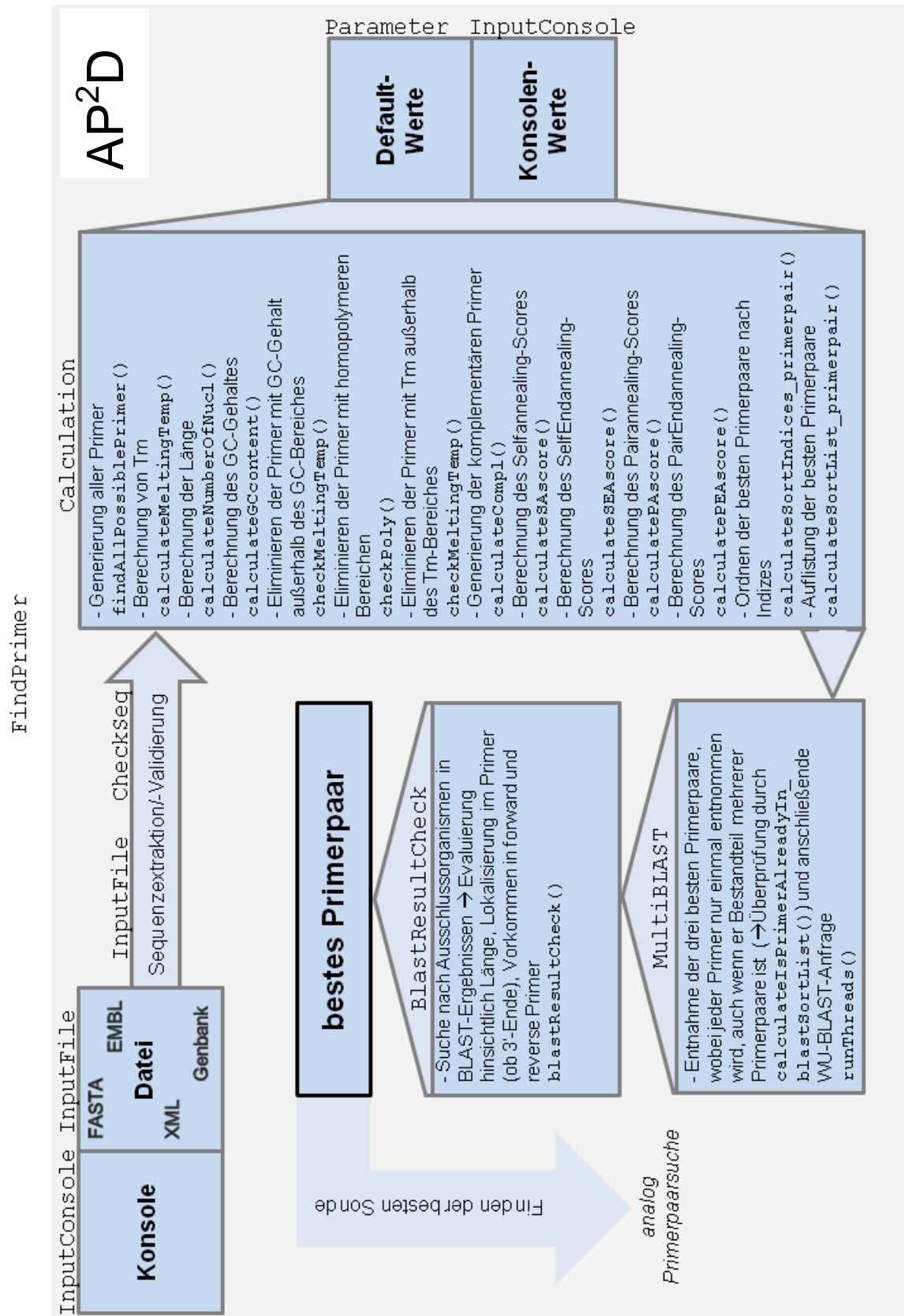
Der Workflow von AP²D ist auf der folgenden Seite zu sehen

5.1 Klassenhierarchie

Der Algorithmus AP²D besteht aus zehn verknüpften Klassen:

- `BlastResultCheck`
- `BlastThread`
- `CalculationPrimer`
- `CheckSeq`
- `InputConsole`
- `InputFile`
- `MultiBlast`
- `Parameter`
- `Probe`
- `RunClient`

Die zwei Calculation-Klassen (`CalculationPrimer`, `Probe`) beinhalten alle Methoden zur Berechnung des besten Primerpaares bzw. der besten Sonde. Diese werden in der Klasse `PrimerSearch` bzw. `Probe` aufgerufen und schließlich ausgeführt. In den Klassen `MultiBlast`, `BlastThread`, `RunClient` und `BlastResultCheck` finden sich Methoden für die Ausführung und Überprüfung der Spezifität. `InputFile` und `InputConsole` sind für die Verarbeitung eingetragener Werte verantwortlich. In der Klasse `CheckSeq` erfolgt die Überprüfung einer eingegebenen Sequenz.



5.2 Sequenzerfassung

Um eine dynamische Eingabe zu ermöglichen, kann der User zwischen einer manuellen Eingabe auf der Kommandozeile und der Verwendung einer eigenen Datei wählen. Die Eingabe verschiedener Sequenzformate wird hierbei unterstützt.

Das **Sequenzformat** ist die Art und Weise, wie die Sequenz in einer Datei hinterlegt wird. [Gaedeke 2007]

Als mögliche Formate für diese Dateien wurden das XML-, das Fasta-, das GenBank und das EMBL-Format gewählt, da diese Formate zu den gebräuchlichsten zählen und in wichtigen biologischen Datenbanken zu finden sind, wie z. B. in der Genbank oder der European Molecular Biology Laboratory (EMBL/EBI) Nukleotide Sequence Database. [Surzycki 2003]

Das Einlesen der Sequenz nach manueller Eingabe erfolgt in die Klasse `InputConsole` durch die Methode `readConsoleInput()`. Der User wird aufgefordert eine Sequenz einzugeben:

```
Please input a sequence or press [return] to open a file chooser.
```

Sollte er dies nicht tun und ohne Eingabe die Enter-Taste betätigen, wird die Methode `gettingInputFile()` der Klasse `InputFile` ausgeführt. Hierdurch kann man mittels eines Datei-Öffnen-Dialogs die entsprechende Datei auswählen. Anschließend erfolgt die zeilenweise Speicherung des Dateiinhaltes in einen `StringBuilder` namens `arr`:

```
while ((s = br.readLine()) != null)
{
    arr.add(s);
}
```

Für diese Aufgabe wurde der `StringBuilder` gewählt, da es sich auch um sehr lange Eingabesequenzen handeln kann und dieser im Gegensatz zu dem *String*-Datentyp wesentlich schneller ist. Veranschaulicht wird dies in Tabelle 5-4. Hierbei wurde das Wort „Wort“ mittels einer Schleife immer wieder gespeichert.

Tabelle 5-4: Vergleich von String und StringBuilder

anhand der Laufzeiten bei unterschiedlich vielen Schleifendurchläufen in denen „Wort“ gespeichert wird

	Anzahl an Schleifendurchläufen				
	100	500	1.000	5.000	10.000
	Laufzeit [ms]				
String	0	5	15	213	716
StringBuilder	0	0	0	1	1

Es ist ersichtlich, dass dies durch den StringBuilder wesentlich effizienter geschieht, als bei einem einfachen String. Durch dessen Verwendung wird somit die Effizienz des Algorithmus' wesentlich gesteigert.

Entsprechend dem verwendeten Sequenzformat erfolgt anschließend die Weiterverarbeitung.

Das **FASTA-Format** (siehe Abbildung 5-8) beinhaltet eine Überschriftzeile und die Sequenz. [Gaedeke 2007]

```
>gi|188497078|dbj|AB437010.1| Candida albicans genes for ITS1, 5.8S rRNA, ITS2, complete sequence, strain: IFM 55382
CTGATTGCTTAATTGCACCACATGTGTTTTCTTTGAAACAACTTGCTTTGGCGGTGGGCCCCAGCCTG
CCGCCAGAGGTCTAAACTTACACCAATTTTTTATCAACTGTGCACCCAGATTATTACTAATAGTCAAA
ACTTTCAACAACGGATCTCTTGGTTCTCGCATCGATGAAGAACGCGAGCGAAATGCGATACGTAATATGAA
TTGCGATATTCGTGAATCATCGAATCTTTGAACGCACATTCGCGCCCTCTGGTATTCGCGAGGGCATGCC
TGTTTGAGCGTCGTTTCTCCCTCAAACCGCTGGGTTTGGTGTGAGCAATACGACTTGGGTTTGCTTGAA
AGACGCTAGTGGTAAGCGGGATCGCTTTGACAAATGGCTTAGGCTTAACCAAAACATTGCTTGGCGCGG
TAACGTCCACCAGTATATCTTCAAAC
```

Abbildung 5-8: FASTA-Format des Genbank-Eintrages 188497078

Hierdurch kann die Speicherung der Sequenz ab der zweiten Zeile erfolgen:

```
if (pfad.endsWith(".fasta")) {
    for (int i = 1; i < arr.size(); i++) {
        for (int r = 0; r < arr.get(i).length(); r++) {
            sb.append(arr.get(i).charAt(r));
        }
    }
    seq = sb.toString().replace(" ", "");
}
```

Ein **Genbank-File** beinhaltet vor der Sequenz eine Fülle an Informationen. Die eigentliche Sequenz beginnt erst nach dem Wort „Origin“ (zu sehen in Abbildung 5-9)

```
Runigami gun, Okinawa, Japan
institution of Itano, E. N.: Laboratory of Applied Immunology,
Department of Pathological Sciences, Center of Biological Sciences,
State University of Londrina, 86051-970, Londrina, Parana, Brazil
institution of Inomata, T.: Department of Experimental Animals,
Faculty of Veterinary Medicine, Azabu University, 1-17-71,
Fuchinobe, Sagami-hara, 229-8501, Kanagawa, Japan.
FEATURES
    source
        1..447
            /organism="Candida albicans"
            /mol_type="genomic DNA"
            /strain="IFM 55382"
            /isolation_source="exhaled breath of male Rough-toothed
            dolphin called Larf"
            /host="Steno bredanensis (Rough-toothed dolphin)"
            /db_xref="taxon:5476"
            /country="Japan:Okinawa"
            /collection_date="Feb-2007"
            /note="experimental number: 207;
            genotype A"
    misc_RNA
        1..138
            /product="internal transcribed spacer 1"
    rRNA
        139..296
            /product="5.8S ribosomal RNA"
    misc_RNA
        297..447
            /product="internal transcribed spacer 2"
ORIGIN
    1 ctgatttgct taattgcacc acatgtgttt ttctttgaaa caaacttgct ttggcggtgg
    61 gccacgctg ccgccagagg tctaaactta caaccaattt ttatcaact tgcacacca
    121 gattattact aatagtcaaa actttcaaca acggatctct tggttctcgc atcgatgaag
    181 aacgcagcga aatgcgatac gtaatatgaa ttgcagatat tcgtgaatca tcgaatcttt
    241 gaacgcacat tgcgcacctt ggtattccgg agggcatgcc tgtttgagcg tcgtttctcc
    301 ctcaaacccg tgggtttggt gttgagcaat acgacttggg ttgcttgaa agacggtagt
    361 ggtaaggcgg gatcgctttg acaatgcctt aggtctaac aaaaacattg ctgctggcgg
    421 taacgtccac cagctatata ttcaaac
//
```

Abbildung 5-9: Genbank-Format des Genbank-Eintrage 188497078

Das heißt, die Speicherung richtet sich nach diesem Wort. Sie beginnt eine Zeile später und endet in der vorletzten Zeile. Anschließend werden alle Leerzeichen

gelöscht. Weiterhin werden mittels eines regulären Ausdrucks alle Zahlen gelöscht:

```
if (pfad.endsWith(".gb")) {
    int counteri = 0;
    for (int i = 0; i < arr.size(); i++) {
        //wenn das wort "ORIGIN" auftritt
        if (arr.get(i).contains("ORIGIN")) {
            //merke es dir mit counteri
            counteri = i;

            //für alle zeilen nach counteri bis zur vorletzten zeile
            for (int k = counteri + 1; k < arr.size()-1; k++) {
                for (int r = 0; r < arr.get(k).length(); r++) {
                    //speichere
                    sb.append(arr.get(k).charAt(r));
                }
            }
        }
    }
    seq = sb.toString().replaceAll(" ", "").replaceAll("[0-9]*", "");
}
```

Wie in dem Genbank-File, sind auch bei dem **XML-Format** sehr viele weitere Informationen zu der Sequenz enthalten. Die Sequenz befindet sich zwischen den Identifiern „<IUPACna>“ und „</IUPACna>“. Das Problem der Sequenzextraktion wurde in diesem Fall dadurch gelöst, dass der String „<IUPACna>“ gesucht wird und anschließend alles eingefügt wird, bis das Zeichen „<“ auftritt.

Das EMBL-Format ist dem Genbank-Format ähnlich, auch hier liegt die Sequenz am Ende des Files und wird durch eine bestimmte Zeichenabfolge („SQ Sequence“) angekündigt. Die Speicherung erfolgt demnach analog zu der eines Genbank-Files.

Durch die Methode `SequenceCheck()` der Klasse `CheckSeq` erfolgt die Evaluierung der eingegeben Sequenz, d. h. sowohl von der aus dem File erfassten, als auch von der mittels Konsole erfassten Sequenz. Entsprechend dem folgenden Pseudocode erfolgt eine Überprüfung nach gültigen Zeichen, deren Häufigkeit und Reihenfolge. Die Eingabesequenz muss hierbei folgendem regulären Ausdruck genügen muss:

$$w = \{A, T, G, C\}^+ + [\{A, T, G, C\}^+ +] + \{A, T, G, C\}^+$$

```

Zeichenkette input = Rückgabewert der Methode readConsoleInput()
Zahlenwert tagFrontCounter1 = 0
Zahlenwert tagBackCounter1 = 0
Zahlenwert tagFrontCounter2 = 0
Zahlenwert tagBackCounter2 = 0
Zahlenwert unKnownChar = 0
boolscher Wert bad = falsch
Zeichenkette stringToCheck= input mit Großbuchstaben

Für i=1→|stringToCheck|
    Falls stringToChecki≠A oder stringToChecki≠C oder
    stringToChecki≠G oder stringToChecki≠T
        erhöhe unKnownChar um eins
    ende falls
    Falls stringToChecki ≙ [
        erhöhe tagFrontCounter1 um eins
    ende falls
    Falls stringToChecki ≙ ]
        erhöhe tagBackCounter 1 um eins
    ende falls
    Falls tagFrontCounter1 ≙ 1 und tagBackCounter1 ≙ 1
        Für h=1→|stringToCheck|
            Falls stringToCheckh=[
                erhöhe tagFrontCounter2 um eins
            ende falls
            Falls stringToCheckh=]
                erhöhe tagBackCounter2 um eins
            ende falls
        ende für
    ende falls
ende für
Falls tagFrontCounter1 ≙ 1 und tagBackCounter1 ≙ 1
und tagBackCounter2 ≙ 0
    Ausgabe: "wrong order: ']' then '['. right order: '[' dann ']'. "
    bad = wahr
ende falls
Falls tagFrontCounter1 ≙ 0
    Ausgabe: "input contains no '['."
    bad = wahr
ende falls
Falls tagBackCounter1 ≙ 0
    Ausgabe: "input contains no ']'. "
    bad = wahr
ende falls
Falls unKnownChar > 1
    Ausgabe: „input contains illegale chars!“
    bad = wahr
ende falls
Falls bad = wahr
    Ausgabe: „program finished. Press [return] to restart]
    rufe diese Methode erneut auf
ende falls
setze alle Zahlenwerte = 0
gib stringToCheck zurück

```

5.3 Default-Werte

Zur Berechnung der Primer bzw. Sonden werden verschiedene Parameter benötigt. Der Benutzer von AP²D hat hierbei die Wahl, ob er einen Wert über die Kommandozeile eingibt oder auf die gespeicherten Default-Werte zurückgreift. Zur Abfrage der Werte erscheint eine Aufforderung in der Konsole, wie z. B. die folgende für die minimale Primerlänge.

```
Please set the minimale primer length [bp]. Press [return] to get the
default-value (18 bp).
```

Durch diese Ausgabe kann der User sofort erkennen welchen Wert der Default-Wert hat und sofern er sich dafür entscheidet, greift das Programm auf die entsprechende Get-Methode der Klasse `Parameter` zu. In dem geschilderten Fall ist dies die Methode `getLengthMin()`.

Die für AP²D gewählten Default-Werte sind in Tabelle 5-5 dargestellt.

Tabelle 5-5: Default-Werte des Programmes

Parameter	Wert	Quelle
minimale Primerlänge	18	[van Pelt-Verkuil 2008] [Mülhardt 2009]
optimale Primerlänge	24	Mittelwert
maximale Primerlänge	30	[van Pelt-Verkuil 2008] [Mülhardt 2009]
minimaler GC-Gehalt Primer	40	[Mülhardt 2009]
optimaler GC-Gehalt Primer	50	[Dieffenbach 2003] [van Pelt-Verkuil 2008]
maximaler GC-Gehalt Primer	60	[Mülhardt 2009]
minimale Schmelztemperatur Primer	50	abgeleitet von [Dieffenbach 2003]
optimale Schmelztemperatur Primer	57	[Dieffenbach 2003]
maximale Schmelztemperatur Primer	65	abgeleitet von [Dieffenbach 2003]
Formamid-Gehalt	0	[Sarker 1990]
Natrium-Gehalt	50	[URL 4]
maximale fwd-rev-Primer-Schmelztemperaturdifferenz	5	[van Pelt-Verkuil 2008]

5.4 Primersuche

5.4.1 Generierung der Primer

`findAllPossiblePrimers()`

Anhand der eingegebenen Sequenz und der eingegebenen Werte für die minimale und maximale Primerlänge erfolgt durch die Methode `findAllPossiblePrimers()` die Generierung aller Primer, die mindestens die minimale und maximal die maximale Primerlänge besitzen. Die Generierung der forward Primer erfolgt vor dem Zeichen [und die der reverse Primer nach dem Zeichen]. Veranschaulicht werden die Bereiche, die zur Generierung der Primer genutzt werden in Abbildung 5-10.

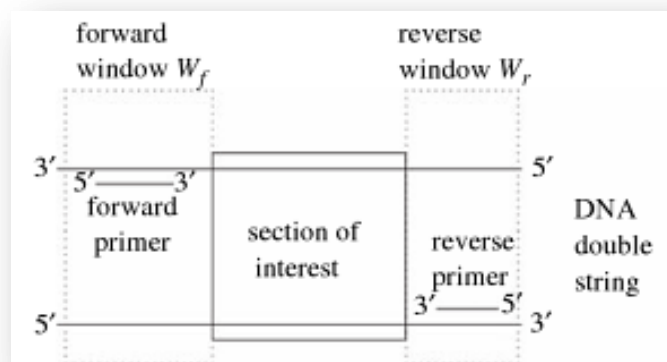


Abbildung 5-10: Darstellung der Bereiche der eingegebenen Sequenz, die zur Generierung der fwd und rev Primer genutzt werden [Kämpke 2001]

Die Generierung der ersten Primer ist vereinfacht in Abbildung 5-11 zu sehen. Die Realisierung erfolgt wie im folgenden Pseudo-Code dargestellt.

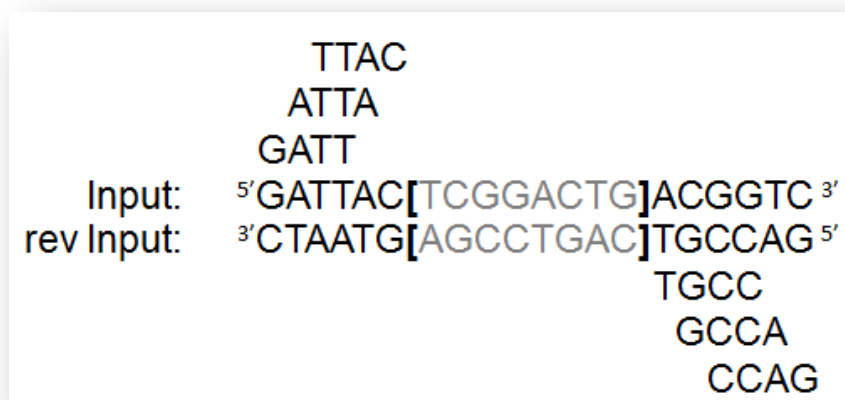


Abbildung 5-11: Generierung aller möglichen Primer der Länge 4 anhand einer vorgegebenen Inputsequenz


```
Übernimm Zeichenkette input
Übernimm Zahlenwert pMinLength
Übernimm Zahlenwert pMaxLength

Initialisiere leere Primerliste pList
Initialisiere Zahlenwert revPrimerBegin = 0
Initialisiere Zeichenkette w = input

für v = 1  $\rightarrow$  |w|
  falls  $w_v == [$ 
    für i = pMinLength  $\rightarrow$  pMaxLength
      für k = 1  $\rightarrow$  ( $w_v - i$ )
         $w^F = \{w_k, w_{k+1}, \dots, w_{k+i}\}$ 
        füge zu pList das Teilwort  $w^F$  hinzu
      ende für
    ende für
  ende falls
  falls  $w_v == ]$ 
    |pList| = revPrimerBegin
    für i = pMinLength  $\rightarrow$  pMaxLength
      für k =  $w_v + 1 \rightarrow (|w| - i)$ 
         $w^R = \{w_{k+i}, w_{k+i-1}, w_{k+i-2}, \dots, w_k\}$ 
        füge zu pList das Teilwort  $w^R$  hinzu
        erstelle dabei, gemäß Watson-
          Crick-Regel die komplementäre Sequenz
      ende für
    ende für
    füge revPrimerBegin zu pList
  ende falls
ende für
```

5.4.2 Chemische Eigenschaften der generierten Primer

Nach Beendigung der Methode `findAllPossiblePrimers()` werden für alle generierten Primer die Schmelztemperatur, die Anzahl der Nukleotide und der GC-Gehalt berechnet. Dies geschieht mittels Methoden der Klasse `Calculate`.

calculateMeltingTemp()

Durch die Methode `calculateMeltingTemp()` erfolgt die Berechnung der Schmelztemperatur für jeden einzelnen Primer gemäß Nearest-Neighbor-Methode (Formel 4). Die Ergebnisse hierzu werden in eine `ArrayList` gespeichert, um sie später wieder abzurufen:

```
Übernimm Primerliste pList
Initialisiere leere Schmelztemperaturliste TmList
Initialisiere leeren Zahlenwert Enthalpie
Initialisiere leeren Zahlenwert Entropie
Initialisiere leeren Zahlenwert TmWert

Für l = 1 → |pList-1|
    w = pListl
    Für m = 1 → |w|
        Initialisiere Zeichenkette zk = {wm, wm+2}

        addiere Werte für Enthalpie und Entropie entspre-
        chend Tabelle 3.1

        TmWert = Berechnung entsprechend Formel (4)
        Füge zu TmList TmWert
        TmWert = 0
        Enthalpie = 0
        Entropie = 0
    ende für
```

calculateNumberOfNucl()

Weiterhin wird eine `ArrayList` mit der Anzahl der Nukleotiden jedes einzelnen Primers erstellt. Dies geschieht durch die Methode `calculateNumberOfNucl()`.

```
Übernimm Liste primerList
Initialisiere leere Nucleotidanzahlliste numberOfNucl
Initialisiere Zahlenwert numberOfNuclValue = 0

Für i = 1 → |primerList-1|
    numberOfNucl = |primerListi|
    Füge numberOfNuclValue zu numberOfNucl
ende für
```

calculateContentOfGC()

Mittels der Methode `calculateContentOfGC()` wird die Berechnung des GC-Gehaltes realisiert. Der GC-Gehalt ergibt sich, wie in Formel 9 zu sehen aus der Division der Anzahl aller Guanin und Cytosin durch die Länge des Primers und der Multiplikation dieses Ergebnisses mit 100 um auf Prozent zu kommen.

$$\%GC = \frac{G+C}{G+C+A+T} \times 100 \quad (9)$$

A	...Adeninanzahl
C	...Cytosinanzahl
G	...Guaninanzahl
T	...Thyminanzahl

Die Speicherung der Werte erfolgt in einer weiteren ArrayList. Im folgenden Pseudocode ist die Umsetzung dessen zu sehen.

```
Übernimm Liste primerList
Übernimm Liste numberOfNucl

Initialisiere leere GCgehaltListe contentOfGC
Initialisiere Zahlenwert numberOfGC = 0

Für i = 1 → |primerList|-1
  wr = primerListi
  Für k = 1 → |wr|
    Falls wkr = G oder wkr = C
      erhöhe numberOfGC um eins
    ende falls
  ende für
  berechne mittels numberOfGC und numberOfGCi den GC-Gehalt und
  füge ihn in contentOfGC ein
  numberOfContent=0
ende für
```

5.4.3 Bewertung der Primer nach ihren chemischen Eigenschaften

Um zu gewährleisten, dass sich die Primer in dem gewünschten Schmelztemperatur- und GC-Gehalt-Bereich befinden, erfolgt anschließend die Eliminierung der Primer, die nicht den gewünschten Kriterien entsprechen. Ebenso werden Primer gelöscht, deren Sequenz mehr als eine vorher definierte Anzahl an Wiederholungen einer Base besitzt, da dies die Spezifität der Primer herabsetzt. In einer weiteren ArrayList wurde die Schmelztemperaturdifferenz der Primerpaare dargestellt.

checkMeltingTemp()

Die Überprüfung, ob sich der Primer in dem gewünschten Schmelztemperaturbereich befindet, erfolgt in der Methode *checkMeltingTemp()* der *Calculator*-Klasse und wird wie im folgenden Pseudocode dargestellt, umgesetzt:

```
Übernimm Liste meltingTemp
Übernimm Zahlenwert pMeltingMin
Übernimm Zahlenwert pMeltingMax
Übernimm Liste primerList
Übernimm Liste numberOfNucl
Übernimm Liste contentOfGC

Initialisiere Zahlenwert revPrimerBegin = primerList|primerList|

Für i = 1 → |pList-1|
    Falls meltingTempi < pMeltingMin | meltingTempi > pMeltingMax
        Entferne meltingTempi, meltingTempi, numberOfNucli, contentOfGCi
        Falls i < revPrimerBegin
            entferne primerList|primerList|
            füge i zu primerList
            Falls revPrimerBegin < 1
                Ausgabe: „no primer pair in melting temperature range possible“
            ende falls
        ende falls
        erniedrige i um eins
    ende falls
ende für
```

Die zu übernehmenden Listen sind jene, welche in vorangegangenen Methoden erzeugt wurden. Die Zahlenwerte werden durch die Methoden *readConsolepLengthMin()* und *readConsolepLengthMax()* übergeben. Durch den Aufruf dieser Methoden besteht die Möglichkeit auf die Default-Werte zurückzugreifen oder eigene Werte zu setzen.

```
Please set the minimale primer length [bp]. Press [return] to get the
default-value (18 bp).

Please set the maximale primer length [bp]. Press [return] to get the
default-value (30 bp).
```

Wenn ein Primer eine Schmelztemperatur besitzt, die nicht in dem angegebenen Bereich liegt, so werden er (aus der ArrayList primerList) und alle dazugehörigen Werte (Eintrag in den Listen meltingTemp, numberOfNucl, contentOfGC) gelöscht. Wenn es sich bei diesem Primer um einen forward Primer handelt, wird der Zahlenwert revPrimerBegin um eins verringert.

checkContentOfGC()

In der Methode `checkContentOfGC()` erfolgt die Überprüfung, ob ein Primer im vorher definierten GC-Gehalt-Bereich liegt. Analog zur Methode `checkMeltingTemp()` werden zunächst die ArrayListen primerList, meltingTemp, contentOfGC, und numberOfNucl übernommen. Die Methode verläuft wie `checkMeltingTemp()`, mit dem Unterschied, dass anstelle von pMeltingMin und pMeltingMax auf pGCMin und pGCMax geprüft wird.

checkPoly()

In der Methode `checkPoly()` wird geprüft, ob in einem Primer eine bestimmte Base mehrfach nacheinander auftritt. Es erfolgt wieder zunächst die Übernahme der ArrayListen. Bei der Methode `checkPoly()` wird auf 4 verschiedene reguläre Ausdrücke geprüft: `[A]{4,}`, `[C]{4,}`, `[G]{4,}`, `[T]{4,}`

Ein **regulärer Ausdruck** dient der Beschreibung eines Musters (engl. Pattern) zur Verarbeitung von Zeichenketten. Durch die Klasse `Pattern` wird der reguläre Ausdruck in eine interne Darstellung kompiliert und kann nun mit Hilfe der Funktion `Pattern.matches()` auf eine Zeichenkette angewendet werden. Über den zurückgegebenen `Matcher` erfolgt dann der anschließende Zugriff auf das Ergebnis. [URL 5]

In der Methode `checkPoly()` wird dies wie folgt umgesetzt:

```
Pattern patternG = Pattern.compile("[G]{4,}");
Pattern patternC = Pattern.compile("[C]{4,}");
Pattern patternA = Pattern.compile("[A]{4,}");
Pattern patternT = Pattern.compile("[T]{4,}");
Matcher mG = patternG.matcher(primer.get(i));
Matcher mC = patternC.matcher(primer.get(i));
Matcher mA = patternA.matcher(primer.get(i));
Matcher mT = patternT.matcher(primer.get(i));
```

Durch `[A]`, `[C]`, `[G]`, `[T]` werden die Zeichenklassen `{A}`, `{C}`, `{G}`, `{T}` definiert. Mittels `{4,}` erfolgt die Definition eines Quantifizierers (Wiederholungsfaktors). Die zuvor definierten Zeichenklassen kommen hierbei mindestens viermal vor. [URL 5]. Sofern eines dieser definierten Ausdrücke in einem Primer vorkommt, verfährt `checkPoly()` wie die Methode `checkContentOfGC()` bei dem Auftreten eines Primers, dessen GC-Gehalt nicht im definierten Bereich liegt.

calculateMeltingDif_primer()

Da es wichtig ist, dass die Primer eine ähnliche Schmelztemperatur besitzen, wird die Differenz aller Primerpaare mittels der Methode

`calculateMeltingDif_primer()` berechnet und in einer ArrayListe gespeichert. Berechnet wird der Unterschied durch den Betrag der Differenz von der Schmelztemperatur des forward Primers zu der des reverse Primers.

5.4.5 Bewertung der Primer nach ihrer Neigung zu Fehlhybridisierungen

Ein entscheidendes Kriterium für die Auswahl eines Primerpaares ist die Neigung Sekundärstrukturen zu formen oder Fehlhybridisierungen mit dem zweiten Primer zu bilden. Beides sollte vermieden werden, um eine effiziente und gewinnbringende Amplifikation zu ermöglichen.

`calculateSAscore()`

Die Tendenz eines Primers zum Self-Annealing (Selbsthybridisierung) wird durch die Methode `calculateSAscore()` beschrieben. Hierbei wird der Primer gegen sich selbst verschoben und im Falle einer Hybridisierung wird der Score (= Zahlenwert) erhöht. Veranschaulicht wird dies in Abbildung 5-12.

Self annealing	SA score
5' - G A T T - 3' 3' - T T A G - 5'	0
5' - G A T T - 3' 3' - T T A G - 5'	0
5' - G A T T - 3' 3' - T T A G - 5'	0
5' - G A T T - 3' 3' - T T A G - 5'	4
5' - G A T T - 3' 3' - T T A G - 5'	4
5' - G A T T - 3' 3' - T T A G - 5'	0
5' - G A T T - 3' 3' - T T A G - 5'	0

Abbildung 5-12: Beispiel zur Berechnung des SA-Scores

Um den Primer danach zu prüfen, erfolgt zunächst in der Methode `calculateCompl_primer()` die Erzeugung einer ArrayList, die den komplementären Primer enthält:

calculateCompl_Primer()

```

Übernimm Liste primerList

Initialisiere leere KomplPrimerListe compl_primer

Für i = 1 → |primerList-1|
    wr = primerListi
    füge zu compl_primer das Teilwort wr hinzu
    ersetze dabei A mit W, C mit X, G mit Y, T mit Z,
        W mit T, X mit G, Y mit C, Z mit A
ende für

```

Die ArrayLists primerList und compl_primer werden danach der Methode calculateSAscore() übergeben. Diese prüft, wie im folgenden Pseudocode dargestellt, ob es zu einer Fehlhybridisierung kommen kann. Durch die Verwendung der komplementären Sequenz gilt nicht A-T, sondern A-A bzw. T-T und nicht G-C, sondern G-G bzw. C-C als Hybridisierung.

```

Übernimm Liste primerList
Übernimm Liste compl_primer

Initialisiere die leere Self-Annealing-Score-Liste saScoreList
Initialisiere Zahlenwert saScoreValue = 0

Für i = 1 → |pList-1|
    wF = primerListi
    wR = compl_primeri
    Für j = 1 → |wF|
        Für l = 1 → |wR|
            Falls wFj = wRl
                falls wRl = G oder wRl = C
                    saScoreValue = saScoreWert + 4
                sonst
                    saScoreValue = saScoreWert + 2
            ende falls/sonst
        ende falls
    ende für
    füge zu saScoreList saScoreValue
    saScoreValue = 0
ende für

```

Bei einer A-T-Hybridisierung (d. h. A-A oder T-T) wird die Zahl zwei zum Score addiert, bei einer G-C-Hybridisierung (d. h. G-G oder C-C) vier. Dies wurde aus dem Grund gewählt, da Guanin und Cytosin über drei Wasserstoffbrückenbindungen miteinander hybridisieren und somit gegenüber Adenin und Thymin, die nur über zwei Wasserstoffbrückenbindungen hybridisieren, eine stärkere Bindung ausbilden.

calculateSEAScore()

Zur Ermittlung des Self-End-Annealing Scores wird die Methode `calculateSEAScore()` genutzt. Hierbei wird abermals ein Primer gegen sich selbst verschoben, jedoch wird hierbei nur darauf geachtet, ob es zu einer Hybridisierung am 3'-Ende kommt (siehe Abbildung 5-13).

Self end annealing	SEA score
5' - G A T T - 3' 3' - T T A G - 5'	0
5' - G A T T - 3' 3' - T T A G - 5'	2
5' - G A T T - 3' 3' - T T A G - 5'	0
5' - G A T T - 3' 3' - T T A G - 5'	0

Abbildung 5-13: Beispiel zur Berechnung des SEA-Scores

Eine Hybridisierung würde analog zu `calculateSEAScore()` zu einer Addition von entweder zwei oder vier führen. Dem 3'-Ende eines Primers kommt eine besondere Bedeutung zu, da hieran die DNA-Polymerase bindet um die Elongation zu beginnen. Ein hoher SEA-Score, d. h. die starke Neigung eines Primers mit seinem 3'-Ende zu hybridisieren, könnte demnach die gesamte Polymerasekettenreaktion unterbinden. Im folgenden Pseudocode wird die Umsetzung dieses Verfahrens in der Methode `calculateSEAScore()` dargestellt:

```

Übernimm Liste primerList
Übernimm Liste compl_primer

Initialisiere die leere Self-End-Annealing-Score-Liste seaScore
Initialisiere Zahlenwert seaScoreValue = 0

Für i = 1 → |pList-1|
  wP = primerListi
  wC = compl_primeri
  Für h = 1 → |wC|
    Für r = 1 → h
      Falls wP|wp|-r = wCh-r
        Falls wP|wp|-r = G oder Falls wP|wp|-r = C
          seaScoreValue = seaScoreWert + 4
        sonst
          seaScoreValue = seaScoreWert + 2
        ende falls/sonst
      ende falls
    ende für
  ende für
  füge seaScoreValue zu seaScore
  seaScoreValue = 0
ende für

```


calculatePAScore()

Die Methode `calculatePAScore()` berechnet den Par-Annealing Score. Dieser wird ähnlich dem SA Score berechnet, mit dem Unterschied, dass hierbei der forward und der reverse Primer gegeneinander verschoben werden. Der folgende Pseudocode veranschaulicht dies.

```

Übernimm Liste primerList
Übernimm Liste compl_primer

Initialisiere die leere Pair-Annealing-Score-Liste paScore
Initialisiere Zahlenwert paScoreValue = 0
Initialisiere Zahlenwert revPrimerBegin = pListpList

Für i = 1 → revPrimerBegin
    Für j = revPrimerBegin + 1 → |compl_primer|
        wP = primerListi
        wC = compl_primerj
        Für k = 1 → |wP|
            Für r = 1 → |wC|
                Falls wPk = wCr
                    Falls wPk = G oder wPk = C
                        paScoreValue = paScoreValue + 4
                    sonst
                        paScoreValue = paScoreValue + 2
                ende falls/sonst
            ende falls
        ende für
    ende für
    füge zu paScore paScoreValue
    paScoreValue = 0
ende für

```

calculatePEAScore()

Die Pair-End-Annealing Score-Berechnung erfolgt durch die Methode `calculatePEAScore()`. Sie erfolgt ähnlich dem SEA Score, jedoch wird hierbei die Tendenz zur Hybridisierung des forward mit dem reverse bzw. des reverse mit dem forward Primer ermittelt. Im folgenden Pseudocode ist die Realisierung zur Berechnung des PEA Scores dargestellt.

```

Übernimm Liste primerList
Übernimm Liste compl_primer

Initialisiere die leere Pair-EndAnnealing-Score-Liste peaScore
Initialisiere Zahlenwert peaScoreValue = 0
Initialisiere Zahlenwert revPrimerBegin = pListpList
Initialisiere boolschen Wert isPrimerLength = false

Für r = 1 → revPrimerBegin
    Für l = revPrimerBegin + 1 → |compl_primer|
        wP = primerListr
        wC = compl_primerl

```

```

    Falls  $|w^p| \geq |w^c|$ 
      Für  $k = 1 \rightarrow |w^p|$ 
        Für  $y = 1 \rightarrow |w^c| - k$ 
          Falls  $w_{wp-y}^p = w_{k+y}^c$ 
            Falls  $w_{|wp|-y}^p = G$  oder  $w_{|wp|-y}^p = C$ 
              paScoreValue = paScoreValue
                + 4
            sonst
              paScoreValue = paScoreValue
                + 2
            ende falls/sonst
          Falls  $y = |w^c|$ 
            isPrimerLength = wahr
          ende falls
        sonst
          brich (Für  $y = 1 \rightarrow |w^c|$ ) ab
        ende falls/sonst
      ende für
      Falls isPrimerLength = wahr
        brich (Für  $k = 1 \rightarrow |w^p| - k$ ) ab
      ende falls
    ende für
  sonst
     $w^p = \text{primerList}_1$ 
     $w^c = \text{compl\_primer}_r$ 
    Für  $k = 1 \rightarrow |w^p|$ 
      Für  $y = 1 \rightarrow |w^p| - k$ 
        Falls  $w_{wp-y}^p = w_{k+y}^c$ 
          Falls  $w_{|wp|-y}^p = G$  oder  $w_{|wp|-y}^p = C$ 
            paScoreValue = paScoreValue
              + 4
          sonst
            paScoreValue = paScoreValue
              + 2
          ende falls/sonst
        Falls  $y = |w^c|$ 
          isPrimerLength = wahr
        ende falls
      sonst
        brich (Für  $y = 1 \rightarrow |w^c|$ ) ab
      ende falls/sonst
    ende für
    Falls isPrimerLength = wahr
      brich (Für  $k = 1 \rightarrow |w^p| - k$ ) ab
    ende falls
  ende für
  füge zu paScore paScoreValue
  paScoreValue = 0
ende für
ende für

```

5.4.6 Bewertung der Primerpaare nach allen Eigenschaften

calculateGeneralScore()

Der General Score stellt eine Bewertung eines beliebigen Primerpaares (p, q) dar. Er wird durch die Methode `calculateGeneralScore()` berechnet. Hierzu werden zunächst die ermittelten Werte zur Länge, Schmelztemperatur, SA Score, SEA Score der Primer p und q bzw. die Werte für den PA Score und PEA Score des Paares (p, q) vektoriell in einem vorläufigen Score zusammengefasst:

$$\text{Score}(\mathbf{p}, \mathbf{q}) = (|p|, |q|, \%GC(p), \%GC(q), Tm(p), Tm(q), SA(p), SA(q), SEA(p), SEA(q), PA(p, q), PEA(p, q)) \quad (10)$$

Anschließend wird die Differenz zum optimalen Score ermittelt (siehe Formel (12)). Der optimale Score beinhaltet alle zu Beginn definierten optimalen Werte. Die Werte für die Fehlhybridisierungen (SA score, SEA score, PA score, PEA score) sind bei dem optimalen Score Null:

$$\text{Optimal score} = (|P_{opt}|, |P_{opt}|, \%GC, \%GC, Tm_{opt}, Tm_{opt}, 0, 0, 0, 0, 0, 0) \quad (11)$$

Zusätzlich erfolgt die Addition von 1000000, sofern die Schmelztemperaturdifferenz höher ist als der Wert für die definierte maximale Schmelztemperaturdifferenz. Die Abfrage der maximalen Schmelztemperaturdifferenz erfolgt durch die Methode `readConsoleMaxMelting()` der Klasse `InputConsole`, d. h. der Benutzer von AP²D hat die Möglichkeit zwischen einem Default-Wert und der manuellen Eingabe zu wählen.

$$\text{General score} = |\text{Optimal Score} - \text{Score}(\mathbf{p}, \mathbf{q})| + (1000000) \quad (12)$$

Die Speicherung jedes General Score erfolgt in einer ArrayList.

5.4.7 Finden des besten Primerpaares

Nach der Evaluierung der einzelnen Primer und der Primerpaare erfolgt die Ermittlung des besten Primerpaares anhand der gegebenen General Scores.

calculateSortIndices_primerpair()

Um aus der ArrayList mit den gespeicherten General Scores das beste Primerpaar zu ermitteln, erfolgt durch die Methode `calculateSortIndices_primerpair()` die Generierung einer ArrayList, die die General Scores geordnet enthält. Aus Formel 12 wird ersichtlich, dass ein Primerpaar umso besser geeignet ist, je näher es am optimalen Score ist und damit der General Score umso kleiner ist. Somit beginnt die geordnete generalScoreListe mit dem kleinsten Wert. Um das dazugehörige Primerpaar grei-

fen zu können erfolgt die Generierung einer ArrayList, die die Indizes der ursprünglichen General Score-Liste entsprechend der sortierten Liste enthält. Durch folgenden Java-Code wird dies realisiert:

```
for (int i = 0; i < generalScoreSort.size(); i++) {
    for (int j = 0; j < generalScore.size(); j++) {
        if (generalScoreSort.get(i) == generalScore.get(j)) {
            primerpairIndices.add(j+1);
        }
    }
}
```

calculateSortList primerpar()

Auf die durch `calculateSortIndices_primerpair()` generierte ArrayList greift die Methode `calculateSortList_primerpair()` zu, um daraus den forward und reverse Primer jedes Indizes zu generieren. Hierzu geschieht zunächst die Berechnung der Anzahl aller reverse Primer. Diese ergibt sich aus der Größe der ArrayList, in der sich alle gespeicherten Primer befinden abzüglich eins, da unter dem letzten Index der Zahlenwert für den `revPrimerBegin`, d. h. die Anzahl der `fwdPrimer`, gespeichert ist.

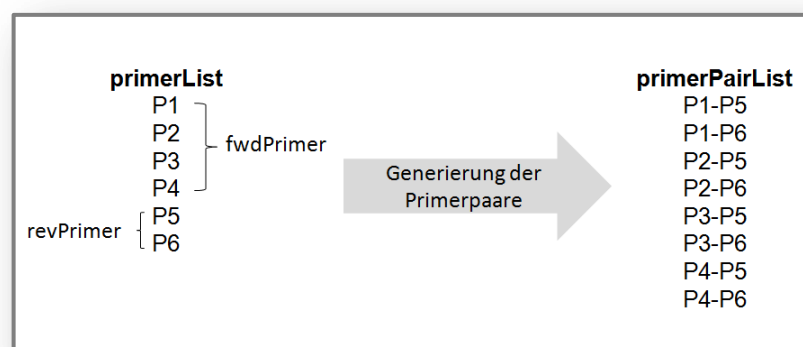


Abbildung 5-14: Generierung einer PrimerpaarListe

Da die Generierung der Primerpaare wie in Abbildung 5-14 geschieht, kann folgender Java-Code verwendet werden, um den forward oder reverse Primer eines beliebigen Indizes zu generieren.

```
for (int i = 1; i < primerpairIndices.size(); i++) {
    int fwdPrimerNumber = primerpairIndices.get(i)
        / (numberOfAllrevPrimer);
    double revPrimerNumber = primerpairIndices.get(i)
        - ((numberOfAllrevPrimer) * fwdPrimerNumber)
        + (revPrimerBegin + 1);
    if ((primerpairIndices.get(i) % (numberOfAllrevPrimer)) != 0)
        fwdPrimerNumber = fwdPrimerNumber + 1;
}
```

5.4.8 Spezifitätsüberprüfung

Die Überprüfung der Spezifität erfolgt mit Hilfe des Programmes BLAST.

BLAST zählt, neben FASTA, zu den bedeutendsten heuristischen Suchalgorithmen in Sequenzdatenbanken. [Böckenhauer 2009] [Xiong 2006] Bei diesen Algorithmen ist nicht gewährleistet, dass sie das optimale Alignment¹⁰ finden, jedoch sind sie 50-100 Mal schneller als die dynamische Programmierung, die darauf ausgerichtet ist. BLAST, als auch FASTA liefern ähnliche Ergebnisse, jedoch liefert FASTA nur ein finales Alignment, wohingegen BLAST mehrere best-scoring Alignments einer Sequenz liefert. Dadurch eignet sich BLAST für die Aufgabe der Spezifitätsüberprüfung besser. [Xiong 2006]

Das Basic Local Alignment Search Tool (BLAST) besteht aus einem Suchalgorithmus und einer Bewertung der gefundenen Lösungen. Der Suchalgorithmus ergibt sich aus drei Einzelschritten:

Zunächst erfolgt die Auflistung aller ähnlichen Teilstrings einer vorher definierten Wortlänge w in der Eingabesequenz und der Datenbank. Für kurze DNA-Sequenzen wird üblicherweise die Wortlänge 11 gewählt. Diese Teilstrings werden als *Hits* bezeichnet. Hierbei sucht man alle lokalen Alignments ohne Lücken, deren Bewertung sich über einem *Schwellenwert* T befindet. T ist dabei abhängig von der verwendeten *Substitutionsmatrix* [Böckenhauer 2009]. Die in dem Nukleotid-BLAST des EBI verwendete Matrix ist die DNA Identity Matrix. Hierbei werden identische Basen positiv und unterschiedliche Basen negativ bewertet (siehe Tabelle 5-6).

Tabelle 5-6: DNA Identity Matrix des EBI [URL 6]

	A	T	G	C
A	1			
T	-10000	1		
G	-10000	-10000	1	
C	-10000	-10000	-10000	1

Anschließend werden von diesen Hits alle Paare gesucht, die höchstens einen *Abstand* d voneinander haben. Dabei ist d ein vorherdefinierter Parameter, welcher von der Länge w abhängig ist. Hits, die zu keinem dieser Paare zuzuordnen sind, bleiben im Weiteren unberücksichtigt.

Nun geschieht eine Ausdehnung der Paare an beiden Enden der Hits. Abgeschlossen ist diese Ausdehnung, wenn sich die Bewertung dadurch nicht mehr erhöht. Wenn sich nach diesem Schritt die Bewertung über dem *Schwellenwert* S befindet, wird das entsprechende Hit-Paar als „high scoring pair“ (*HSP*) bezeichnet. Beginnend mit der besten Bewertung erfolgt nun die Ausgabe der HSPs.

¹⁰ Zuordnung von Entsprechungen in verschiedenen Sequenzen [Dehmer 2006]

Neben dieser Bewertung wird auch ein sogenannter *Bit-Score* erstellt, welcher den Vergleich mit verschiedenen Berechnungen ermöglicht, da er unabhängig von der verwendeten Bewertungsmatrix ist. Weiterhin erfolgt eine Abschätzung der statistischen Signifikanz der gefundenen Ergebnisse. [Böckenhauer 2009]

Um innerhalb des Programmes Zugriff auf BLAST zu ermöglichen, wurde ein BLAST-WebService integriert. Es wurde der BLAST-Webservice des European Molecular Biology Laboratory (EMBL-EBI) gewählt, da die Java-Implementierung hierbei am besten zu realisieren war. Das EMBL-EBI stellt das europäische Pendant zum NCBI da und nutzt die EMBL Nucleotide Database. Diese Datenbank gehört der Vereinigung „International Nucleotide Sequence Database Collaboration“ an. Genbank, DDBJ (DNA Data Base of Japan) und die EMBL Nucleotide Database aktualisieren ihre Einträge aller 24 h, indem sie ihre Neuzugänge den anderen Mitgliedern zugänglich machen. Somit besitzen diese drei Datenbanken einen identischen Sequenzpool. [Gaedeke 2007]

Bei dem BLAST des EBI handelt es sich um WU-Blast, welches von Warren Gishan der Washingtoner Universität aus dem NCBI BLAST 1.4 entwickelt wurde. [Gibas 2001] WU BLAST gilt als schneller und sensibler als das NCBI-BLAST. [Pevser 2009]

Um ein Aufrufen des BLAST-Webservice zu ermöglichen, wurde die Main-Methode der Klasse `WUBlastClient` modifiziert:

```
public void runClient(String sequence,String emailAddress, String outfile)
{
    String statement = null;
    if(outfile == null){
        statement = "-D em_rel_sts -p blastn -E 1000 -b 500 -S
high -m identity --stype dna " + sequence + " --email "
+ emailAddress;
    }
    else{
        statement = "-D em_rel_sts -p blastn -E 1000 -b 500 -S
high -m identity --stype dna " + sequence + " --email "
+ emailAddress + " --outfile " + outfile;
    }

    String[] args = statement.split(" ");
}
```

Als Datenbank (-D) wurde die Standard-Datenbank des EMBL-EBIs für Nukleotide gewählt: `em_rel_sts`.

Das zu nutzende Blastprogramm (-p) ist `blastn`, da es sich bei der Eingabesequenz um Nukleotide handelt.

Als *expect threshold*, *E* (-E) wurde der höchstmögliche Wert 1.000 genutzt. Unter dem *E*-Wert versteht man die Anzahl an Hits mit einem entsprechenden Score, der bei der Sequenzsuche in einer Datenbank zu erwarten ist. Je niedriger er ist, umso höher ist die Wahrscheinlichkeit, dass es sich nicht nur um eine zufällige Übereinstimmung handelt. Für kurze Sequenzen sollte der *E*-Wert jedoch stark erhöht werden, da kurze Sequenzen sehr oft zufällig in der Datenbank gefunden

werden. Um jedoch überhaupt Ergebnisse zu erzielen, muss dieses „Rauschen“ ermöglicht werden. [Gaedeke 2007]

Für die maximale Anzahl der Alignments (-b) wird der Maximalwert 500 gewählt, um möglichst viele Ergebnisse der Suche abzufangen.

Die Sensitivität (-S) wurde auf hoch gesetzt um nur Sequenzen zu finden, die nahezu komplett mit der Eingabesequenz übereinstimmen.

Die genutzte Substitutionsmatrix (-m) ist die oben beschriebene Identity-Matrix.

Durch -stype wird die Eingabesequenz beschrieben. Bei der Suche nach Primern ist dies DNA.

In der Methode `runThreads()` der Klasse `MultiBLAST` erfolgt die Übergabe der Liste mit den geordneten Primerpaaren.

isPrimerAlreadyIn_blastSortList()

Da jeder Primer Bestandteil mehrerer Primerpaare ist und bei der BLAST-Anfrage nicht mehrmals die gleiche Primersequenz gesucht werden soll, muss zunächst eine Überprüfung stattfinden, wann ein Primer zum ersten Mal auftritt. Dies geschieht durch die Methode `isPrimerAlreadyIn_blastSortList()` der `Calculation`-Klasse.

```
Übernimm Liste sortList_primerpair

Initialisiere die leere Liste firstTime_primer
Initialisiere Zahlenwert firstTime = 0

Für i = 1 → |sortList_primerpair|
    Für r = 1 → i+1
        Falls sortList_primerpairr = sortList_primerpairi
            firstTime = r
            füge firstTime zu firstTime_primer hinzu
            brich (Für r = 1 → i+1) ab
        ende falls
    ende für
ende für
```

Die `ArrayList` mit den Werten für das erste Auftreten eines Primers werden ebenfalls der Methode `runThreads()` übergeben. Die Generierung der Werte für `sequence`, `emailAdress` und `outfile` erfolgt durch folgende Schleife:

```
for(int i = 0; i < nr_to_blast;i++){
    if(firstTime_primer.get(i)<i)
        continue;
    else
        sequence=sortList_primerpair.get(i);

    //immer wieder neue Instanzierung der Klasse BlastThread für jeden
    Primer
    bt = new BlastThread();
```

```
        bt.setEmail("jbrettsc@htwm.de");
        bt.setSequence(sequence);
        bt.setOutFile("P_" + i);

        bt.startThread();
    }
```

Durch die Neuinstanziierung je Iterationsschritt sowie den Einsatz von Multi-Threading-Techniken wird erreicht, dass mehrere BLAST-Anfragen zeitgleich gestellt werden können, was eine deutliche Zeiteinsparung zur Folge hat. Die Parameter `sequence`, `emailAdress` und `outfile` werden der Klasse `BlastThread` übergeben. Anschließend erfolgt in dieser Klasse der Aufruf der `start` Methode der `BlastThread` internen Thread-Instanz, wodurch die Anfrage mit Hilfe des EBI-WebClients ausgelöst wird.

5.4.9 Auswertung der BLAST-Suche

Die Auswertung der BLAST-Anfrage erfolgt durch die Methode `blastResultCheck()` der Klasse `BlastResultCheck`.

blastResultCheck()

Zunächst erfolgt die Übergabe der Liste mit den Indizes zur Angabe, wann ein Primer zum ersten Mal Teil eines Primerpaares ist und der sortierten Liste mit allen Primerpaaren. Anschließend wird ermittelt, welche Text-Files ausgewertet werden sollen. Dies geschieht wie folgt:

```
first:for (int r = 0; r < firstTime_primer.size(); r += 2) {
    String line, name = "P_" + firstTime_primer.get(r) + ".out.txt";
    File fwdPrimer = new File(name);
    String line1, name1 = "P_" + firstTime_primer.get(r + 1)
                        + ".out.txt";
    File revPrimer = new File(name1);
    while (!(fwdPrimer.exists()) && !(revPrimer.exists())) {
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

Da eine BLAST-Anfrage je nach Serverauslastung und Query-Sequenz unterschiedlich viel Zeit benötigt, wird vor der Auswertung in regelmäßigen Abständen (hier: alle 5 s) überprüft, ob beide Sequenzen existieren. Falls dies nicht der Fall sein sollte, wird solange gewartet, bis beide existieren.

Zur Evaluierung der Spezifität eines Primers wurde die Auswertung des durch die BLAST-Anfrage generierten `out.txt`-Files gewählt, da in dieser Datei alle nötigen

Informationen enthalten sind. Abbildung 5-15 zeigt den Ausschnitt eines solchen Files.

```
>EM_FUN:AY208049 AY208049.1 Candida albicans strain BCRC 20513 internal
transcribed spacer 1, complete sequence.
Length = 139

Plus Strand HSPs:

Score = 18 (36.4 bits), Expect = 1.2, P = 0.70
Identities = 18/18 (100%), Positives = 18/18 (100%), Strand = Plus / Plus

Query:      1 CCAGAGGTCCTAAACTTAC 18
             |||
Sbjct:      74 CCAGAGGTCCTAAACTTAC 91

>EM_FUN:AY208051 AY208051.1 Candida albicans strain BCRC 21538 internal
transcribed spacer 1, complete sequence.
Length = 139

Plus Strand HSPs:

Score = 18 (36.4 bits), Expect = 1.2, P = 0.70
Identities = 18/18 (100%), Positives = 18/18 (100%), Strand = Plus / Plus

Query:      1 CCAGAGGTCCTAAACTTAC 18
             |||
Sbjct:      74 CCAGAGGTCCTAAACTTAC 91

>EM_FUN:Y14002 Y14002.1 Candida albicans ITS1 DNA, strain HD-H372
Length = 139

Plus Strand HSPs:

Score = 18 (36.4 bits), Expect = 1.2, P = 0.70
Identities = 18/18 (100%), Positives = 18/18 (100%), Strand = Plus / Plus

Query:      1 CCAGAGGTCCTAAACTTAC 18
             |||
Sbjct:      74 CCAGAGGTCCTAAACTTAC 91

>EM_FUN:AY207067 AY207067.1 Candida albicans strain BCRC 20512 internal
transcribed spacer 1, complete sequence.
Length = 140
```

Abbildung 5-15: Ausschnitt einer mittels des BLAST-Webservice generierten out.txt-Datei

Es ist ersichtlich, dass die Nennung des Organismus, in welchem eine ähnliche Sequenz gefunden wurde, in der Zeile steht, die mit dem Zeichen „>“ beginnt. Die dazu gehörende Sequenz befindet sich in der Zeile, welche mit „Sbjct: “ beginnt. Der Inhalt, der mit „>“ beginnenden Zeile, wird vollständig in einer Liste gespeichert. Dies geschieht für jeden Primer in einer separaten Liste. Wenn eine Zeile mit „Sbjct: “ gefunden wurde, wird nach folgendem regulären Ausdruck gesucht: „^Query:\\W*[0-9]+\\W*([ATCG]+)\\W*[0-9]+\$“. Bei dessen Auftreten erfolgt die Lokalisierung der Sequenz durch den Aufruf der `group()`-Methode, welche die Position der Zeichenkette angibt, die `([ATGC]+)` entspricht. Nun erfolgt die Speicherung der Sequenz:

Auf Basis der hierdurch generierten Liste wird nun nach den Ausschlussorganismen gesucht. Exemplarisch soll dies durch die Suche nach einer Sequenz von *Homo sapiens* verdeutlicht werden:

```
content_fwd = Liste mit den extrahierten Werten für den fwdPrimer
Zahlenwert homo_sapiensTest = 0

Für i = 1 → |content_fwd|
  Falls homo_sapiensTest = 0 und content_fwdi enthält "Homo sapiens"
    Für b = 1 → |content_fwd| && content_fwdb beginnt nicht mit ">"
      String testSeq = content_fwdb
      Falls Primer mit testSeq endet
        Falls |testSeq| > 7
          homo_sapiensTest = 1
        ende falls
      ende falls
    ende für
  ende falls
ende für
```

Analog dazu erfolgt die Überprüfung nach *Saccharomyces cerevisiae*, *Penicillium spp.*, *Aspergillus spp.*, *Cladosporium spp.*, *Aureobasidium spp.*, *Alternaria spp.* und *Fusarium spp.* Die Werte für diese Überprüfung werden von beiden Primern in separaten Listen gespeichert. Anschließend erfolgt die abschließende Bewertung. Es wird überprüft, ob bei beiden Primern die gleiche Fehlhybridisierung auftritt, sollte dies der Fall sein wird in die Liste specificity_primerpair die Zahl zwei gespeichert. Kommt nur in einem der Primer eine Fehlhybridisierung zustande, wird die Zahl eins eingefügt und bei gar keiner Fehlhybridisierung die Zahl null. Da dies entsprechend einer speziellen Anordnung erfolgt (*Homo sapiens*, *Saccharomyces cerevisiae*, *Penicillium spp.*, *Aspergillus spp.*, *Cladosporium spp.*, *Aureobasidium spp.*, *Alternaria spp.* und *Fusarium spp.*), ist die Zuordnung zu dem entsprechenden Organismus möglich. Wenn die Zahl eins oder zwei mehr als einmal in der Liste specificity_primerpair auftritt, wird das Primerpaar verworfen und es beginnt die Auswertung der BLAST-Ergebnisse des nächst besseren Primerpaares. Das Auftreten von einer Fehlhybridisierung ist legitim, da man im Anschluss bei der Sondengenerierung hierauf achten kann, dass explizit diese Fehlhybridisierung nicht auftritt.

5.5 Generierung der Sonden

Im Anschluss an die Primergenerierung kann auf Basis dessen eine Generierung von Sonden zum Design eines Microarrays folgen.

5.5.1 Eingabe

Die Werte für fwd-Primer, revPrimer und Eingabesequenz erfolgen in der Klasse *Probe*. Aus der zu Beginn eingegebenen Sequenz und der Sequenz für forward

und reverse Primer wird die Target-Sequenz generiert. Dies geschieht wie folgend dargestellt in der Main-Methode der Klasse `Probe`:

```
fwdPrimerBegin = seq.indexOf(primer_fwd);
isFwdPrimerFound = fwdPrimerBegin != -1;
if (isFwdPrimerFound == true)
    revPrimerEnd = seq.indexOf(primer_rev_fwd)
        + primer_rev_fwd.length();
revPrimerFound = revPrimerEnd != primer_rev_fwd.length() - 1;
if (revPrimerFound == true)
    target = seq.substring(fwdPrimerBegin, revPrimerEnd).replace("[",
        "").replace("]", "");
```

Vor diesem Schritt erfolgt die Generierung des reverse Komplement des reverse Primers, so dass dieser in der Eingabesequenz zu finden ist.

5.5.2 Generierung der Sonden

Die Generierung der Sonden erfolgt in der ebenfalls in der Main-Methode der Klasse `Probe` und wird wie im folgenden Pseudocode dargestellt, realisiert.

```
Übernimm generierte Zeichenkette target
Übernimm Zahlenwert probeMinLength
Übernimm Zahlenwert probeMaxLength

Initialisiere leere Primerliste probeList
Initialisiere Zeichenkette w = probeList

für v = 1 → |w|
    für i = probeMinLength → probeMaxLength
        für k = 1 → (w_v - i)
            wF = {w_k, w_{k+1}, ..., w_{k+i}}
            füge zu probeList das Teilwort wF hinzu
            Falls |wF| = |target| oder
            target endet mit wF und wF = probeLengthMax
                v = |w|
            ende falls
        ende für
    ende für
ende für
```

5.5.3 Chemische Eigenschaften der generierten Sonden

Um eine adäquate Bewertung der Sonden vorzunehmen, erfolgt die Kalkulation der chemischen Eigenschaften für jede Sonde.

calculateMeltingTemp_probe()

Die Berechnung der Schmelztemperatur erfolgt in der Methode `calculateMeltingTempProbe()` der `Calculation`-Klasse. Hierbei wird zwi-

schen zwei Fällen unterschieden: Sonden ≤ 40 Nukleotide und Sonden > 40 Nukleotide.

Bei Ersterem erfolgt die Berechnung nach Breslauer, Formel (1), analog zur Berechnung der Schmelztemperatur für Primer. Bei Sonden mit einer Länge von über 40 Nukleotiden erfolgt die Berechnung nach Meinkoth, Formel (9).

calculateNumberOfNucl_probe()/calculateContentOfGC_probe()

In den Methoden:

`calculateNumberOfNucl_probe()/calculateContentOfGC_probe()`

erfolgt analog zu:

`calculateNumberOfNucl_primer()/calculateContentOfGC_primer()`

die Berechnung der Länge bzw. des GC-Gehaltes der Sonden.

5.5.4 Bewertung der Sonden nach ihren chemischen Eigenschaften

Anhand der berechneten Werte für die Eigenschaften erfolgt die Auslese von Sonden, die den eingegebenen Kriterien für den Schmelztemperatur- und GC-Gehaltbereich. Ebenso wird überprüft, ob Sonden existieren, die eine mehrfache Wiederholung einer bestimmten Base besitzen. Sollte dies der Fall sein, erfolgt die Löschung der Sonde und aller dazugehörigen Werte.

checkMeltingTemp_probe()

In der Calculation-Methode `checkMeltingTemp_probe()` erfolgt die Löschung von Sonden und den dazugehörigen Werten, bei Über- oder Unterschreitung der definierten Schmelztemperaturbereichen. Dies wird wie folgt realisiert:

```
for (int i = 0; i < meltingTemp.size(); i++) {
    if (meltingTemp.get(i) < pMeltingMin
        || meltingTemp.get(i) > pMeltingMax) {
        primer.remove(primer.get(i));
        numberOfNucl.remove(numberOfNucl.get(i));
        meltingTemp.remove(meltingTemp.get(i));
        contentOfGC.remove(contentOfGC.get(i));
        i--;
    }
}
```

checkPoly_probe()

In der Methode `checkPoly_probe()` erfolgt analog zu `checkPoly_primer()` die Löschung von Sonden, die eine mehrfache Wiederholung einer Base besitzen, mit dem Unterschied, dass kein `revPrimerBegin` existiert.

checkContentOfGC_probe()

In der Methode `checkContentOfGC_probe()` erfolgt analog zu `checkMeltingTemp_probe()` die Auslese von Sonden, die in dem definierten GC-Bereich befindlich sind.

5.5.5 Neigung der Sonden zu Fehlhybridisierungen

Wie bei den Primern, so stellt auch bei den Sonden, die Tendenz Fehlhybridisierungen zu bilden ein entscheidendes Kriterium für die Auswahl der besten Sonde dar. Nach der Ausführung der Methode `calculateCompl_probe()` erfolgt in den Methoden `calculateSAscore_probe()` und `calculateSEAscoreProbe()` analog zu den Methoden der Primer die Bewertung der Tendenzen. Dies geschieht unter der Berücksichtigung, dass kein Zahlenwert `revPrimerBegin` existiert, da keine reverse Sonden in der Liste gespeichert sind.

5.5.6 Bewertung der Sonden nach allen Eigenschaften

Die Erzeugung des `GeneralScore` der Sonden erfolgt in der Methode `calculateGeneralScore_probe()` analog zu `calculateGeneralScore_primer()`, unter der Berücksichtigung, dass keine reverse Primer existieren und keine Berechnung zum Pair-Annealing –und Pair-End-Annealing erfolgte.

5.5.7 Finden der besten Sonde

Um die beste Sonde aus der Liste der Sonden zu extrahieren wird wie zur Berechnung des besten Primerpaares die Ausführung zweier Methoden benötigt: `calculateSortIndices_probe()` und `calculateSortList_probe()` der `Calculation`-Klasse. Hierbei erfolgt zunächst die Erzeugung einer `ArrayList`, die die geordneten Indizes der Sonden beinhaltet und anschließend kann mittels des ersten Indizes dieser Liste die beste Sonde gewonnen werden.

5.5.8 Spezifitätsüberprüfung

Die Spezifitätsüberprüfung erfolgt mittels einer BLAST-Anfrage. Hierzu muss in der Klasse `RunClient` die Sequenz der besten Sonde eingetragen werden und dieselbe muss ausgeführt werden. Die Überprüfung der BLAST-Resultate erfolgt manuell.

6. Mittels AP²D erzeugte Primer und Sonden

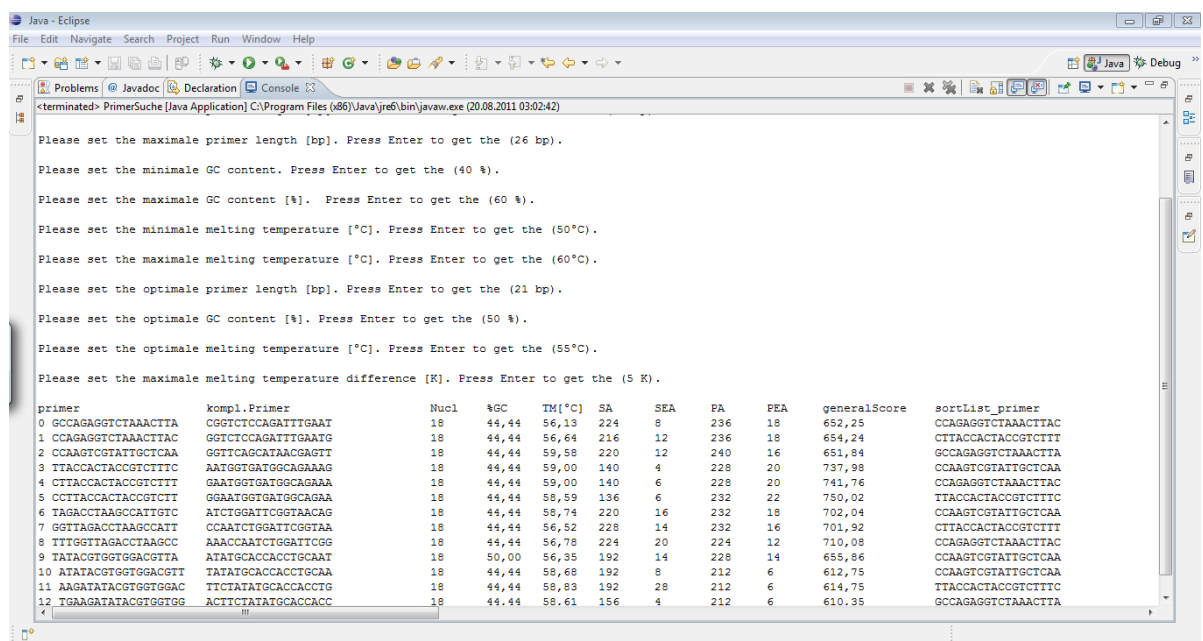
Zur Generierung der Primer und Sonden wurden die von Doreen Kropp erzeugten Sequenzen der rDNA genutzt.

6.1 *Candida albicans*

Die für die Erzeugung der Primer bzw. Sonden genutzte Sequenz ist wie folgt dargestellt. Die hellblauen Abschnitte deklarieren die ITS1- und ITS2-Regionen. Da diese, wie in Kapitel 1.4 beschrieben, zur Generierung der Primer genutzt werden sollen, erfolgt mittels „[„ und „]“ die Eingrenzung der 5,8 S-rDNA.

```
CTGATTGCTTAATTGCACCACATGTGTTTTTCTTTGAAACAAACTTGCTTTGGCGGTGGGCCCAGCCTGCCG
CCAGAGGTCTAAACTTACAACCAATTTTTTATCAACTTGTACACCAGATTATTACTAATAGTCAA [AACTTT
CAACAACGGATCTCTTGGTTCTCGCATCGATGAAGAACGCAGCGAAATGCGATACGTAATATGAATTGCAGAT
ATTCTGTAATCATCGAATCTTTGAACGCACATTGCGCCCTCTGGTATTCCGGAGGGCATGCCTGTTTGAGCGT
CGTTT] CTCCCTCAAACCGCTGGGTTTGGTGTGAGCAATACGACTTGGGTTTGCTTGAAAGACGGTAGTGGT
AAGGCGGGATCGCTTTGACAATGGCTTAGGTCTAACCAAAAACATTGCTTGCGGCGGTAACGTCCACCACGTA
TATCTTCAAAC
```

Unter Verwendung der Default-Werte kam es zur Erzeugung der in Abbildung 6-16 dargestellten Liste.



primer	kompl.Primer	Nucl	%GC	TM[°C]	SA	SEA	PA	PEA	generalScore	sortList_primer
0 GCCAGAGGTCTAAACTTA	CGGTCTCCAGATTGAAT	18	44,44	56,13	224	8	236	18	652,25	CCAGAGGTCTAAACTTAC
1 CCAGAGGTCTAAACTTAC	GGTCTCCAGATTGAATG	18	44,44	56,64	216	12	236	18	654,24	CTTACCACTACCGTCTTT
2 CCAAGTGTATTGCTCAA	GGTTGAGCATAACGAGTT	18	44,44	59,58	220	12	240	16	651,84	GCCAGAGGTCTAAACTTA
3 TTACCACTACCGTCTTT	AATGGTGATGGCAGAAAG	18	44,44	59,00	140	4	228	20	737,98	CCAAGTGTATTGCTCAA
4 CTTACCACTACCGTCTTT	GGAATGGTGATGGCAGAAA	18	44,44	59,00	140	6	228	20	741,76	CCAGAGGTCTAAACTTAC
5 CTTACCACTACCGTCTTT	GGATGGTGATGGCAGAAA	18	44,44	58,59	136	6	232	22	750,02	TTACCACTACCGTCTTT
6 TAGACCTAAGCCATTGTC	ATCTGGATTGGTAACAG	18	44,44	58,74	220	16	232	18	702,04	CCAAGTGTATTGCTCAA
7 GGTTAGACCTAAGCCATT	CCAACTCTGGATTGGTAA	18	44,44	56,52	228	14	232	16	701,92	CTTACCACTACCGTCTTT
8 TTTGGTTAGACCTAAGCC	AAACCAATCTGGAATCGG	18	44,44	56,78	224	20	224	12	710,08	CCAGAGGTCTAAACTTAC
9 TATACGTGGTGGACGTTA	ATATGCAACACCTGCAAT	18	50,00	56,35	192	14	228	14	655,86	CCAAGTGTATTGCTCAA
10 ATATACGTGGTGGACGTT	TATATGCAACACCTGCAAA	18	44,44	56,58	192	8	212	6	612,75	CCAAGTGTATTGCTCAA
11 AAGATATACGTGGTGGAC	TTCTATATGCAACACCTG	18	44,44	58,83	192	28	212	6	614,75	TTACCACTACCGTCTTT
12 TGAAGATATACGTGGTGG	ACTTCTATATGCAACACC	18	44,44	58,61	156	4	212	6	610,35	GCCAGAGGTCTAAACTTA

Abbildung 6-16: Eclipse-Screenshot zum Primer-Design von *C. albicans*

Die dadurch ermittelten Primer und deren dazugehörige Länge, Schmelztemperatur und GC-Gehalt sind in Tabelle 6-7 zu sehen

Tabelle 6-7: Primerpaar für *Candida albicans*

	Primersequence	Schmelztemperatur	Länge	GC-Gehalt
forward	CCAGAGGTCTAAACTTAC	43 °C	18 bp	44 %
reverse	CTTACCACTACCGTCTTT	45 °C	18 bp	44 %

Dies führte zur Bildung folgender Target-Sequenz (die roten Bereiche kennzeichnen die Primer):

CCAGAGGTCTAAACTTACAACCAATTTTTTATCAACTTGTACACCAGATTATTACTAATAGTCAAACTTTC
AACAAACGGATCTCTTGGTTCTCGCATCGATGAAGAACGCAGCGAAATGCGATACGTAATATGAATTGCAGATA
TTCGTGAATCATCGAATCTTTGAACGCACATTGCGCCCTCTGGTATTCCGGAGGGCATGCCTGTTTGAGCGTC
GTTTCTCCCTCAAACCGCTGGGTTTGGTGTGAGCAATACGACTTGGGTTTGCTTGAAAGACGGTAGTGGTAA
G

Die Target-Sequenz besitzt eine Länge von 293 bp, einen GC-Gehalt von 44 % und eine Schmelztemperatur von 94 °C. Weiterhin ist bei diesem Primerpaar mittels BLAST ausgeschlossen wurden, dass eine Fehlhybridisierung mit den Ausschlussorganismen vorkommt.

Die hierfür generierte Sonde ist folgende:

5´-TGGTGTGAGCAATACGACTTGGGTTTGCTTGAAAGACGGTAGTGGAAG-3´

Sie ist 50 bp lang, hat eine Schmelztemperatur von 81 °C und einen GC-Gehalt von 46 %. Da schon die Primer keine Fehlhybridisierungen zulassen, ist es unerheblich, ob die Sonde Hybridisierungen mit den Ausschlusssequenzen ermöglichen würde. Eine Überprüfung ergab jedoch, dass in der generierten Sonde auch keine Fehlhybridisierungen auftreten würden.

6.2 *Candida glabrata*

Zur Generierung der Primer bzw. Sonden wurde die ITS1-5,8S-ITSII-Region genutzt. Für die Primer galt nur der Bereich der Internal transcribed spacer. Folgende Sequenz diente als Eingabesequenz für das Programm.

AAGAAATTTAATTGATTTGTCTGAGCTCGGAGAGAGACATCTCTGGGGAGGACCAGTGTAGACACTCAGGAGG
CTCCTAAATATTTTCTCTGCTGTGAATGCTATTTCTCCTGCGCTTAAGTGCAGCGTTGGTGGGTGTTT
TGCAGTGGGGGAGGGAGCCGACAAAGACCTGGGAGTGTGCGTGGATCTCTCTATTCCAAAGGAGGTGTTTTA
TCACACGACTCGACACTTTCTAATTACTACACAGTGGAGTTTACTTTACTACTATTCTTTTGTTCGTTGGG
GGAACGCTCTCTTTGCGGGGGGAGTTCTCCAGTGGATGCAAACACAAACAAATATTTTTTTAAACTAATTCA
GTCAACACAAGATTTCTTTTAGTAGAAAACAACCTTCA [AACTTTCAACAATGGATCTCTTGGTTCTCGCATC
GATGAAGAACGCAGCGAAATGCGATACGTAATGTGAATTGCAGAATTCGATGATCATCGAATCTTTGAACGC
ACATTGCGCCCTCTGGTATTCCGGGGGGCATGCCTGTTTGAGCGTCATTT] CCTTCTCAAACACATTGTGTTT
GGTAGTGAGTGATACTCGTTTTTGAGTTAACTTGAAATTGTAGGCCATATCAGTATGTGGGACACGAGCGCAA
GCTTCTCTATTAATCTGCTGCTCGTTTGCGCGAGCGGCGGGGGTTAATACTGTATTAGGTTTTACCAACTCGG
TGTTGATCTAGGGAGGGATAAGTGAGTGTTTTGTGCGTGTGGGCAGACAGACGTCCTTTAA

Unter Verwendung der Default-Werte wurde das in Tabelle 6-8 dargestellte Primerpaar ermittelt.

Tabelle 6-8: Primerpaar für *Cancica glabrata*

	Primersequence	Schmelztemperatur	Länge	GC-Gehalt
forward	CGTGGATCTCTCTATTCC	46 °C	18 bp	50 %
reverse	CGAGTATCACTCACTACC	46 °C	18 bp	50 %

Die daraus resultierende Targetsequenz ist die folgend dargestellte Sequenz.

CGTGGATCTCTCTATTCCAAAGGAGGTGTTTTATCACACGACTCGACACTTTCTAATTACTACACACAGTGGA
GTTTACTTTACTACTATTCTTTTGTTCGTTGGGGGAACGCTCTCTTTTCGGGGGGGAGTTCTCCAGTGATGC
AAACACAAACAAATATTTTTTAACTAATTCAGTCAACACAAGATTTCTTTTAGTAGAAAAACAACCTCAAAA
CTTTCAACAATGGATCTCTTGGTTCTCGCATCGATGAAGAACGCAGCGAAATGCGATACGTAATGTGAATTGC
AGAATTCGTGAATCATCGAATCTTTGAACGCACATTGCGCCCTCTGGTATTCCGGGGGGCATGCCTGTTTGA
GCGTCATTTCTCTCTCAAACACATTGTGTTTGGTAGTGAGTGATACTCG

Sie hat eine Länge von 414 bp, eine Schmelztemperatur von 94,1 °C und einen GC-Gehalt von 42 %. Zudem sind bei den Primern keine Fehlhybridisierung mit den Ausschlusssequenzen möglich.

Die aus dieser Sequenz abgeleitet Sonde ist die folgende:

5'-AGCGTCATTTCTCTCTCAAACACATTGTGTTTGGTAGTGAGTGATACTCG-3'

Die Länge der Sonde beträgt 50 bp, sie besitzt eine Schmelztemperatur von 80 °C und ihr GC-Gehalt beträgt 42 %. Mittels BLAST wurde nachgewiesen, dass keine Fehlhybridisierung mit den Ausschlussorganismen vorkommt.

6.3 *Candida guilliermondii*

In der nachfolgenden Sequenz wird die Eingabesequenz zur Generierung der Primer bzw. Sonden für *Candida guilliermondii* gezeigt.

CAGTATTCTTTTGCCAGCGCTTAAGTGCAGCGCGGAAAAACCTTACACACAGTGCTTTTTGATACAGAACTCT
TGCTTTGGTTTGGCCTAGAGATAGGTTGGGCCAGAGGTTTAAACAAAACACAATTTAATTATTTTTACAGTTAG
TCAAATTTTGAATTAATCTTCAA [AACTTTCAACAACGGATCTCTTGGTTCTCGCATCGATGAAGAACGCAGC
GAAATGCGATAAGTAATATGAATTGCAGATTTTCGTGAATCATCGAATCTTTGAACGCACATTGCGCCCTCTG
GTATTCAGAGGGCATGCCTGTTTGAGCGT] CATTCTCTCTCAAACCCCGGGTTTGGTATTGAGTGATACT
CTTAGTCGGACTAGGCGTTTGCTTGAAGTATTGGCATGGGTAGTACT
NGATAGTGCTGTCGACCTCTCAATGTATTAGTTTATCCAACCTCGTTGAATGGTGTGGCGGGATATTTCTGGT
ATTGTTGGCCCGGCCTTACAACAACCAACAAG

Die daraus generierten Primer sind in Tabelle 6-9 zu sehen.

Tabelle 6-9: Primerpaar für *Candida guilliermondii*

	Primersequence	Schmelztemperatur	Länge	GC-Gehalt
forward	CCTTACACACAGTGTCTT	45 °C	18 bp	44 %
reverse	CTACCCATGCCAATACTT	45 °C	18 bp	44 %

Die folgende, daraus resultierende Sequenz besitzt eine Länge von 369 bp, eine Schmelztemperatur von 93 °C und einen GC-Gehalt von 41 %. Des Weiteren wurde mittels BLAST nachgewiesen, dass es bei diesem Primerpaar zu keiner Fehlhybridisierung mit den Ausschlusssequenzen kommen kann.

CCTTACACACAGTGTCTTTTTGATACAGAACTCTTGCTTTGGTTTGGCCTAGAGATAGGTTGGGCCAGAGGTT
TAACAAAACACAATTTAATTATTTTTACAGTTAGTCAAATTTGAATTAATCTTCAAACCTTTCAACAACGGA
TCTCTTGGTTCTCGCATCGATGAAGAACGCAGCGAAATGCGATAAGTAATATGAATTGCAGATTTTCGTGAAT
CATCGAATCTTTGAACGCACATTGCGCCCTCTGGTATTCCAGAGGGCATGCCTGTTTGAGCGTCATTTCTCTC
TCAAACCCCGGGTTTGGTATTGAGTGATACTCTTAGTCGGACTAGGCGTTTGCTTGAA**AAGTATTGGCATGG**
GTAG

Daraus ergab sich folgende Sequenz als Sonde:

5'-GTGATACTCTTAGTCGGACTAGGCGTTTGCTTGAAAAGTATTGGCATGGG-3'

Die Länge der Sonde beträgt 50 bp, sie besitzt eine Schmelztemperatur von 81 °C und einen GC-Gehalt von 46 %. Des weiteren wurde mittels BLAST nachgewiesen, dass keine Fehlhybridisierungen mit den Ausschlusssequenzen auftreten.

6.4 *Candida krusei*

Zur Ermittlung der Primer und Sonden wurde der vollständige ITS1-5,8S-ITS2-Bereich genutzt:

CTGTGATTTAGTACTACTGCGTGAGCGGAACGAAAACAACAACACCTAAAATGTGGAATATAGCATATAGT
CGACAAGAGAAATCTACGAAAAACAACAA [AACTTTCAACAACGGATCTCTTGGTTCTCGCATCGATGAAGA
GCGCAGCGAAATGCGATACCTAGTGTGAATTGCAGCCATCGTGAATCATCGAGTTCTTGAACGCACATTGCGC
CCCTCGGCATTCGGGGGGGCATGCCTGTTTGAGCGT] **CGTTTCCATCTTGCGCGTGCGCAGAGTTGGGGGAGC**
GGAGCGGACGACGTGTAAAGAGCGTCGGAGCTGCGACTGCCTGAAAGGGAGCGAAGCTGGCCGAGCGAACTA
GACTTTTTTTTTCAGGGACGCTTGGCGGCCGAGAGCGAGTGTTCGAGACAACAAAAGCTCGACCTCAAATCAG
GTAGGAATACCCGCTGAACCTAAGCATATCAATAAGCGGAGGAAAAAG

Durch Verwendung der Default-Werte kam es zur Selektion der folgenden Primer als bestes Primerpaar.

Tabelle 6-10: Primerpaar für *Candida krusei*

	Primersequence	Schmelztemperatur	Länge	GC-Gehalt
forward	GCATATAGTCGACAAGAG	45 °C	18 bp	44 %
reverse	TATGCTTAAGTTCAGCGG	48 °C	18 bp	44 %

Die Targetsequenz besitzt eine Länge von 400 bp, eine Schmelztemperatur von 98 °C und einen GC-Gehalt von 53 %. Bei den verwendeten Primern wurde eine Fehlhybridisierung mit den Ausschlusssequenzen mittels BLAST ausgeschlossen.

GCATATAGTCGACAAGAGAAATCTACGAAAAACAAACAAACTTTCAACAACGGATCTCTTGGTTCTCGCATC
GATGAAGAGCGCAGCGAAATGCGATACCTAGTGTGAATTGCAGCCATCGTGAATCATCGAGTTCTTGAACGCA
CATTGCGCCCCCTCGGCATTCCGGGGGGCATGCCTGTTTGAGCGTCGTTTCCATCTTGC GCGTGC GAGAGTTG
GGGAGCGGAGCGGACGACGTGTAAAGAGCGTCGGAGCTGCGACTCGCCTGAAAGGGAGCGAAGCTGGCCGAG
CGAACTAGACTTTTTTTTCAGGGACGCTTGGCGGCCGAGAGCGAGTGTTCGAGACAACAAAAAGCTCGACCTC
AAATCAGGTAGGAATACCCGCTGAACTTAAGCATA

Aus dieser Targetsequenz ergab sich folgende Sonde:

5'-GTGTTGCGAGACAACAAAAAGCTCGACCTCAAATCAGGTAGGAATACCCG-3'

Sie besitzt eine Länge von 50 bp, eine Schmelztemperatur von 82 °C und einen GC-Gehalt von 48 %. Die BLAST-Anfrage ergab, dass es hierbei zu einer Fehlhybridisierung mit *Saccharomyces spp.* kommen kann.

6.5 *Candida parapsilosis*

Der vollständige ITS I-5,8S-ITS II-Bereich wird zur Generierung der Primer und Sonden genutzt.

CAGAATGAAAAGTGCTTAAGTGCATTTTTTCTTACACATGTGTTTTTCTTTTTTTGAAAACCTTTGCTTTGGTA
GGCCTTCTATATGGGGCCTGCCAGAGATTAACTCAACCAATTTTATTTAATGTCAACCGATTATTTAATAG
TCAA [AACTTTCAACAACGGATCTCTTGGTTCTCGCATCGATGAAGAACGCGAAATGCGATAAGTAATAT
GAATTGCAGATATTTCGTGAATCATCGAATCTTTGAACGCACATTGCGCCCTTTGGTATTCCAAAGGGCATGCC
TGTTTGAGCGTCATTT] CTCCCTCAAACCCTCGGGTTTGGTGTGAGCGATACGCTGGGTTTGCTTGAAAGAA
AGGCGGAGTATAAACTAATGGATAGGTTTTTCCACTCATTGGTACAACTCCAAACTTCTTCCAAA

Da bei der Verwendung der Default-Werte kein Ergebnis hervorkam, wurden ein paar Parameter verändert: der minimale GC-Gehalt wurde auf 25 °C gesetzt und der Aufruf der Methode `checkPoly_primer()` wurde unterlassen.

Tabelle 6-11: Primerpaar für *Candida parapsilosis*

	Primersequence	Schmelztemperatur	Länge	GC-Gehalt
forward	CACATGTGTTTTCTTTT	40 °C	18 bp	28 %
reverse	GTACCAATGAGTGGAAAA	42 °C	18 bp	39 %

TGTGTTTTTCTTTTTTTGAAAACCTTTGCTTTGGTAGGCCTTCTATATGGGGCCTGCCAGAGATTAACTCAAC
CAAATTTTATTTAATGTCAACCGATTATTTAATAGTCAAACTTTCAACAACGGATCTCTTGGTTCTCGCATC
GATGAAGAACGCGAGCGAAATGCGATAAGTAATATGAATTGCAGATATTTCGTGAATCATCGAATCTTTGAACGC
ACATTGCGCCCTTTGGTATTCCAAAGGGCATGCCTGTTTGAGCGTCATTTCTCCCTCAAACCCTCGGGTTTGG
TGTGAGCGATACGCTGGGTTTGCTTGAAAGAAAGGCGGAGTATAAACTAATGGATAGGTTTTCCTCACTCAT
TGGTAC

Sie besitzt eine Länge von 317 bp, eine Schmelztemperatur von 93 °C und einen GC-Gehalt von 40 %.

Die Primer wurden so gewählt, dass keine Fehlhybridisierung mit einem der Ausschlussorganismen auftritt. Aus dem Target ergab sich folgende Sonde.

5'-AGAAAGGCGGAGTATAAACTAATGGATAGGTTTTTTTCCACTCATTGGTAC-3'

Da die Target-Sequenz einen sehr hohen A-T-Anteil hat, ergab sich für die Sonde auch nur ein GC-Gehalt von 38 %, eine Schmelztemperatur von 78 °C und eine Länge von 50 bp. Mittels BLAST-Anfrage wurde festgestellt, dass die Sonde keine Fehlhybridisierungen mit den Ausschlussorganismen hat

6.6 *Candida tropicalis*

Die nachfolgende Sequenz (ITSI-5,8S-ITSII-Bereich) wurde als Input zur Generierung der Primer und Sonden genutzt.

CTGATTTGCTTAATTGCACCACATGTGTTTTTTATTGAACAAATTTCTTTGGTGGCGGGAGCAATCCTACCGC
CAGAGGTTATAACTAAACCAAACCTTTTTATTACAGTCAAACCTTGATTTATTATTACAATAGTCAA [ACTTTC
AACAACGGATCTCTTGGTTCTCGCATCGATGAAGAACGCAGCGAAATGCGATACGTAATATGAATTGCAGATA
TTCGTGAATCATCGAATCTTTGAACGCACATTGCGCCCTTTGGTATTCCAAAGGGCATGCCTGTTTGAGCGT]
CATTTCTCCCTCAAACCCCGGGTTTGGTGTGAGCAATACGCTAGGTTTGTGTTGAAAGAATTTAACGTGGAA
ACTTATTTTAAGCGACTTAGGTTTATCCAAAACGCTTATTTTGCTAGTGGCCACCACAATTTATTTTCATAAC

Tabelle 6-12: Primerpaar für *Candida tropicalis*

	Primersequence	Schmelztemperatur	Länge	GC-Gehalt
forward	CGCCAGAGGTTATAACTA	46 °C	18 bp	44 %
reverse	GCCACTAGCAAAATAAGC	46 °C	18 bp	44 %

Die mittels dieser Primer gebildete Target-Sequenz besitzt eine Länge 344 bp, eine Schmelztemperatur von 92 °C und einen GC-Gehalt von 40 %. Bei der Verwendung der oben beschriebenen Primern wurden Fehlhybridisierungen mit den Ausschlusssequenzen mittels BLAST ausgeschlossen.

CGCCAGAGGTTATAACTA AACCAAACTTTTTATTACAGTCAAACCTTGATTTATTATTACAATAGTCAAACCTT
TCAACAACGGATCTCTTGGTTCTCGCATCGATGAAGAACGCAGCGAAATGCGATACGTAATATGAATTGCAGA
TATTCGTGAATCATCGAATCTTTGAACGCACATTGCGCCCTTTGGTATTCCAAAGGGCATGCCTGTTTGAGCG
TCATTTCTCCCTCAAACCCCGGGTTTGGTGTGAGCAATACGCTAGGTTTGTGTTGAAAGAATTTAACGTGGA
AACTTATTTTAAGCGACTTAGGTTTATCCAAAACGCTTATTTTGCTAGTGGC

Die aus dieser Sequenz generierte Sonde ist folgende:

5'-CTTATTTTAAGCGACTTAGGTTTATCCAAAACGCTTATTTTGCTAGTGGC-3'

Sie ist 50bp lang, besitzt einen GC-Gehalt von 36 % und eine Schmelztemperatur von 77 °C. Fehlhybridisierungen wurden mittels BLAST ausgeschlossen.

7. Zusammenfassung und Ausblick

Es zeigte sich, dass alle evaluierten Primer-Programme Schwächen aufweisen. Hierdurch war die Entwicklung eines eigens für die gestellten Anforderungen entsprechenden Programmes unabdingbar. Es entstand ein Programm, welches Funktionen zur Primer- und zur Sonden-Generierung ausführen kann. Eine Besonderheit ist hierbei die Auswertung der Spezifitätskontrolle, d. h. der durch BLAST erzeugten Werte. AP²D ist darauf ausgerichtet, Primer zu generieren, die zuvor definierten Ausschlussorganismen (*Homo sapiens*, *Saccharomyces cerevisiae*, *Penicillium spp.*, *Aspergillus spp.*, *Cladosporium spp.*, *Aureobasidium spp.*, *Alternaria spp.*, *Fusarium spp.*) zu suchen und es sich, im Falle des Auftretens zu merken. Um das Programm benutzerfreundlicher zu gestalten, sollte die Erstellung einer grafischen Benutzeroberfläche erfolgen. Wie in Kapitel 3 beschrieben, ist es sinnvoll, Primer nach ähnlichen Werten für ΔG zu wählen. Diesen Ansatz könnte man verfolgen und somit die Bewertung der Primerpaare verbessern. Die vollständige Automatisierung des Programmes, d. h. dass man keine Änderungen innerhalb des Source-Code vornehmen muss, stellt ebenso eine Möglichkeit zur Erhöhung der Benutzerfreundlichkeit dar. Hierzu wurden schon Ansätze erarbeitet, die in den Java-Dateien als Kommentare zu lesen sind. Die Ausgabe der Ergebnisse könnte als Datei erfolgen. Hierfür wurde die Klasse `OutputFileWrite` erstellt, die nur noch mit den entsprechenden Übergangsparametern vervollständigt werden muss. Sofern ein Primerpaar mit einer Fehlhybridisierung mit den Ausschlussorganismen generiert wird, könnte sich AP²D den entsprechenden Organismus merken und die Sonde so generieren, dass explizit diese Fehlhybridisierung vermieden wird. Die Parameter zur BLAST-Anfrage sollten sich entsprechend der Query-Sequenz anpassen oder vom Benutzer eingeben lassen können. Die Implementierung des NCBI-BLAST bzw. der darauf folgende Vergleich mit WU-BLAST ist ebenfalls ein möglicher Ausblick. Im Anschluss an die Auswertung der BLAST-Dateien, könnte die Löschung derer erfolgen. Die Klasse `DeleteClass` ist hierfür schon vorbereitet wurden. Eine Visualisierung der Selbst- und Paarhybridisierung ließe konkrete Rückschlüsse auf die Hybridisierung zu. Die Verarbeitung weiterer Sequenzformate könnten ebenfalls implementiert werden um das Programm universeller zu gestalten. Die Auswahl, ob eine Primer-, Sonden- oder Primer/Sondengenerierung erfolgen soll, würde das Anwendungsspektrum erhöhen. Hierzu sind in den Klassen `PrimerSuche` und `InputConsole` erste Vorkehrungen getroffen wurden. Die Ein- bzw. Austragung von Ausschlusssequenzen stellt eine weitere Möglichkeit dar, AP²D für die verschiedensten Aufgaben nutzbar zu gestalten. Durch eine Optimierung der einzelnen Methoden könnte eine Effizienzerhöhung erfolgen. Trotz all dieser Entwick-

lungsmöglichkeiten stellt AP²D bereits jetzt ein effizientes Programm zur Generierung von Oligonukleotiden zur Verwendung als Primer und Sonden dar. Die automatische, auf den Anforderungen des Benutzers abgestimmte Spezifitätsüberprüfung ist, wie bei der Recherche nach Internetprogrammen festgestellt, einzigartig. Abschließend ist zu vermerken, dass das gestellte Ziel, die Generierung von Primern und Sonden, erreicht wurde. Zusätzlich wurde ein Programm entwickelt, welches augenscheinlich einzigartig in Hinblick auf die Spezifitätsüberprüfung ist.

Literatur

- [Abd-Elsalam 2003] Abd-Elsalam, K.A.: "Bioinformatic tools and guideline for PCR primer design". African Journal of Biotechnology, 2(2003)5, S.91-95
- [Ahmed 2007] Ahmed, N.; Dawson, M. et al.: "Biology of disease".-1.Aufl. Trowbridge: Cornwell Press, 2007, S.56f
- [Ahsen 2001] Ahsen, N.; Wittwer, C.T.; Schütz, Ekkehard: „Oligonucleotide Melting Temperatures under PCR Conditions: Nearest-Neighbor Corrections for Mg^{2+} , Deoxynucleotide Triphosphate, and Dimethyl Sulfoxide Concentrations with Comparison to Alternative Empirical Formulas".-Clinical Chemistry. (2001)47, S.1956-1961
- [Antranikian 2006] Antranikian, G.: „Angewandte Mikrobiologie".-1.Aufl. Berlin Heidelberg: Springer-Verlag, 2006, S. 216
- [Arora 2009] Arora, K., Berlin, Kück, U: Nowrousian, M; et al.; „Schimmelpilze: Lebensweise, Nutzen, Schaden, Bekämpfung".-3.Aufl. Heidelberg: Springer-Verlag, 2009, S. 10
- [Böckenhauer 2009] Böckenhauer, H.-J.; Bongartz, D.: „Algorithmische Grundlagen der Bioinformatik: Modelle, Methoden und Komplexität".1.Aufl.:2009; S.97-98
- [Brakhage 1990] Brakhage, A.; Jahn, B.; Schmidt, A.: „Aspergillus fumigatis: biology, clinical aspects, and molecular approaches".-2.Aufl. Basel: Karger AG, 1999, S. 193
- [Breslauer 1986] Breslauer, K. J.; Frank, R. et al.: "Predicting DNA duplex stability from the base sequence". *Proc. Natl. Acad. Sci. -USA*: (1986)83, S.3746–3750
- [Bruhn 2008] Bruhn, H.D.; Fölsch, U.R.; Schäfer, H.: „LaborMedizin: Indikationen, Methodik und Laborwerte. Pathophysiologie und Klinik".-2.Aufl. Stuttgart: Schattauer GmbH, 2008, S. 282
- [Busch 2010] Busch, U.: „Molekularbiologische Methoden in der Lebensmittelanalytik: Grundlegende Methoden und Anwendungen".-1.Aufl. Berlin Heidelberg: Springer-Verlag, 2010, S.208-210
- [Buselmaier 2006] Buselmaier, W.; Tariverdia, G.: „Humangenetik für Biologen".-5.Aufl. Heidelberg: Springer-Verlag, 2006, S. 34-25

- [Buselmaier 2006] Buselmaier, W.; Tariverdia, G.: „Humangenetik für Biologen“. -5.Aufl. Heidelberg: Springer-Verlag, 2006, S. 34-25
- [Campbell 1998] Campbell, N.: „Biologie“. -1. Aufl. Heidelberg Berlin: Spektrum Akademischer Verlag, 1998, S. 626 ff.
- [Caspary 2006] Caspary, W.; Kist, M.; Stein, J.: „Infektiologie des Gastrointestinaltraktes“. -5.Aufl. Heidelberg: Springer Verlag, 2006, S. 52
- [Clark 2009] Clark, D.P.; Pazdernik, N.J.: „Molekulare Biotechnologie: Grundlagen und Anwendungen“. -1.Aufl. Heidelberg: Spektrum Akademischer Verlag, 2009, S.106ff
- [De Renaldis 2006] De Rinaldis, E.; Lahm, Ar.: “DNA Microarrays: Current Applications”. -1.Aufl. Wymondham: Horizon Bioscience, 2006, S.2-5
- [Deb 2003] Deb, S.; Deb, S.P.: “Methods in Molecular Biology: p53 protocols”. -1.Aufl. Towana: Humana Press, 2003, S.79
- [Dehmer 2006] Dehmer, M.: „Strukturelle Analyse Web-basierter Dokumente“. -1.Aufl. Wiesbaden: Deutscher Universitäts-Verlag, 2006, S.106
- [Dieffenbach, 2003] Dieffenbach, C.; Dveksler, G.: PCR Primer: A laboratory manual. - 2.Aufl. New York: Cold Spring Harbor, 2003, S. 62-64
- [Dorak 2007] Doral.M.T.: „Real-time PCR“. -1.Aufl. New York, Abington: Taylor & Francis Group, 2007, S.94ff
- [Dresdener Transferbrief 2011] Dresdener Transferbrief, 2.11, 19.jahrgang, ausgabe 3.2011, dresden: progressmedia Verlag & Werbeagentur GmbH, Wricke, E.; Brandl, P.; Kedzierski, U.; Ermisch, B.-V. „Biotype Diagnostic GmbH steht für innovative molekulare Diagnostik“, S.19
- [Gabrio 2001] Gabrio, T.: „Schimmelpilze in Innenräumen – Nachweis, Bewertung, Qualitätsmanagment“. Landes Gesundheitsamt Baden-Württemberg Berichte. – Stuttgart: Landes Gesundheitsamt. – Jahrgang (2001)
- [Gaedeke 2007] Gaedeke, N.: „Biowissenschaftlich recherchieren: Über den Einsatz von Datenbanken und anderen Ressourcen der Bioinformatik“. -1.Aufl. Basel: Birkhäuser Verlag, 2007 a.) S.43-45
- [Gibas 2001] [aus „Developing bioinformatics computer skills“ Cynthia Gibas, Per Jambeck, 2001 O'Reilly&Associates, Inc., Sebastopool, 1.Auflage,S.183]
- [Graw 1995] Graw, J.: „Genetik“. -4. Aufl. Berlin, Heidelberg: Springer Verlag, 2006 a. S.285-290, b. S.54-56

- [Griffin 1994] Griffin, D.: "fungal physiology".-2.Aufl. New York: Willey –Liss Inc., 1994, S. 239
- [Hahn 2004] Hahn, H.; Miksits, K.: „Basiswissen- Medizinische Mikrobiologie und Infektiologie“-3.Aufl.Berlin, Heidelberg: Springer Verlag, 2004, S.216f
- [Hermey 2010] Hermey, G.; Schwake, M.; Sommer-Böchl, T.:„Der Experimentator: Neurowissenschaften“ .-1.Aufl. Heidelberg: Spektrum Akademischer Verlag, 2010, S.17 ff
- [Hirsch-Kaufmann 2009] Hirsch-Kaufmann, M.; Schweiger, M.: „Biologie für Mediziner und Naturwissenschaftler“.-6. Aufl. Stuttgart: Georg Thieme Verlag, 2009, S.63
- [Horton 2008] Horton, R.; Moran, L.: "Biochemie"-4.Aufl. München: Pearson Studium, 2008:a.) S.6ff, b.) S.19ff
- [Kämpke 2001] Kämpke, T.; Kieninger, M.; Mecklenburg, M.: "Efficient primer design algorithms". BIOINFORMATICS.- 17(2001)3, S. 214-225
- [Kennedy 2011] Kennedy, S.; Oswald, N.: "PCR Troubleshooting and Optimization: The Essential Guide".-1.Aufl. Norfolk: Caister Academic Press, 2011, a.)S.112 ff, b.) S.84f
- [Kimmel 2006] Kimmel, A.; Oliver, B.: "DNA microarrays: Array platforms and wet-bench protocols".-1.Aufl. Oxford: Elsevier Inc., 2006, S.73-79
- [Lahoud 2008] Lahoud, G; Timoshchuk, V.; et al.: "Properties of pseudo-complementary DNA substituted with weakly pairing analogs of guanine or cytosine". *Nucl. Acids Res.* 36 (2008)22, S. 6999-7008
- [Linz 1990] Linz, U.; Degenhardt, H.: „Die Polymerase-Kettenreaktion: Ein Überblick“ Berlin-Heidelberg:Springer-Verlag, 1990, S.515- 530
- [Meigel 2010] Meigel, W.; Schöfer, H.: „Infektionen der Haut – Grundlagen, Diagnostik und Therapiekonzepte für Dermatologen, Internisten und Pädiater“-3.Aufl. Stuttgart: Georg Thieme Verlag, 2010, S.170
- [Meinkoth 1984] Meinkoth J.; Wahl G.: "Hybridization of nucleic acids immobilized on solid supports". *Analytische Biochemie*, (1984)2, S.267-284
- [Montera 2008] Montera, L; Nicoletti, M.; Rutawski, L.; et al.: "The PCR Primer Design as a Metaheuristic Search Process".-1.Aufl. Berlin Heidelberg: Verlag, 2008, S. 963–973

- [Mühlhardt 2009] Mühlhardt, C.: Der Experimentator: Molekularbiologie/Genomics.–6.Aufl. Heidelberg: Spektrum Akademischer Verlag, 2009, S. 84-118
- [Müller 2004] Müller, H.-J.; Röder, T.: „Der Experimentator: Microarrays“.- 1.Aufl. München: Spektrum Akademischer Verlag, 2004, S.1-15
- [Pestana 2010] Pestana, E.A.; Belak, S.; Diallo, A. et al.: „Early, Rapid and Sensitive Veterinary Molecular Diagnostics - Real Time PCR“.1-Aufl.Dordrecht:SpringerScience+Busines Media B.V., 2010, S. 9-18
- [Pevser 2009] [Bioinformatics and functional genomics Jonathan Pevser S 144, 2. Ausgabe, wiley blackwell verlag, 2009 Ort:Hoboken, New Jersey
- [Pruesse 2007] Pruesse, E.; Quast, C.; et al.:“SILVA: a comprehensive online resource für quality checked and aligned ribosomal RNA sequence data compatible with ARB”.-Nucleic Acids Research, 25(2007)21; S.7188-7196
- [Regenass-Klotz 2005] Regenass-Klotz, M.: „Grundzüge der Gentechnik: Theorie und Praxis“.- 3.Aufl. Basel: Birkhäuser Verlag AG, 2005 S.13ff
- [Reinhart 2010] Reinhart K., Brunkhorst F.M.; Bone, H.-G. et al.: Prävention, Diagnos, Therapie und Nachsorge der Sepsis“. 1.Revision der S-2k Leitlinien der Deutschen Sepsis-Gesellschaft e.V. (DSG) und der Deutschen Interdisziplinären Vereinigung für Intensiv- und Notfallmedizin (DIVI) 2010, S.5
- [Renneberg 2010] Renneberg, R.; Süßbier, D.:Biotechnologie für Einsteiger.-3.Aufl. Heidelberg:Spektrum Akademischer Verlag, 2010
- [Rychlik 1989] Rychlik, W.; Rhoads, E.: “A computer program for choosing optimal oligonucleotides for filter hybridization, sequencing and in vitro amplification of DNA”. Nucleic Acids Research.-17(1989)21
- [Saikia 2008] Saikia, R.; Bezbaruah, R.; Bora, T.C.:”Microbial Biotechnology“.-1.Aufl. Pitam Pura, Neu Delhi: New India Publishing Agency, 2008, S.1ff
- [SantaLucia 1998] SantaLucia, J., Jr.: “A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics”. *Proc. Natl. Acad. Sci. U. S. A.* (1998)95, S.1460–1465
- [Sarker 1990] Sarker, G.; Kapelner, S.; Sommer, S.S.: „Formamid can dramatically improve the specificity of PCR“.-Nucleic Acids Research, 18(1990)24, S.7465
- [Schmuck 2008] Schmuck, C.; Engels, B. et al: „Chemie für Mediziner“.-1.Aufl. München: Pearson Studium, 2008; S.696ff

- [Strandhagen 2010] Strandhagen, U.: „Weiterentwicklung eines DNS-Microarrays zur Identifikation von humanpathogenen Pilzen“. -2010.-92 S. Tübingen, Universität Tübingen, Institut für Medizinische Mikrobiologie und Hygiene, Dissertation, 2010
- [Surzychi 2003] Surzychi, S.: “Human molecular biology laboratory manual”. -1.Aufl. Cornwall: Blackwell Science Ltd, 2003, S.139-143
- [Tusnady 2005] Tusnady, G.; Simon, I.: “BiSearch: primer-design and search tool for PCR on bisulfite-treated genome”. Nucleic Acids Research. -33 (2005)1
- [van Pelt-Verkuil 2008] van Pelt-Verkuil, E.; van Belkum, Alex; Hays, J.: “Principles and technical aspects of PCR amplification”. - 9.Aufl. Berlin Heidelberg: Springer Verlag, 2008
- [Werdan 2005] Werdan, K; Schuster, H.-P.: „Sepsis und MODS“-4.Aufl. Heidelberg: Springer Medizin Verlag, 2005, S.4ff
- [Werner 2002] Werner, S.: „Untersuchung zur Regulation und Bedeutung der Chitin-Synthese beim Maispathogen *Colletotrichum graminicola* (CES.) WILS. (Teleomorph: *Glomerella graminicola* POLITIS)“. -2002.-124 S. Halle-Wittenberg, Martin-Luther-Universität, Mathematisch-Naturwissenschaftlich-Technische Fakultät, Dissertation, 2002
- [Wolf 1997] Wolf, K.; Böhm, K.: „Organisation von Mikrotubuli in der Zelle. In: Biologie in unserer Zeit“. -Weinheim: Wiley-VCH Verlag 27(1997)2, S. 87–95
- [Wrobel 2004] Wrobel, G.; Kokocinski, F; Lichter, P.: “AutoPrime: selecting primers for expressed sequences”. Genome Biology. -5(2004)11
- [Wu 2004] Wu, L.; Thompson, D.K.: “Development and Evaluation of Microarray-Based Whole-Genome Hybridization for Detection of Microorganisms within the Context of Environmental Applications”. - Environ. Sci. Technol. 2004(38), S.6775-6782
- [Xiong 2006] Xiong, J.: “Essential bioinformatics”. 1.Aufl. Cambridge: University Press, 2006, S 52-62
- [Yuryev 2007] Yuryev, A.: PCR Primer Design. -1.Aufl. Totowa: Humana Press Inc., 2007, S.18ff
- [Zvelibil 2008] Zvelebil, M.; Baum, J.O.: „Understanding Bioinformatics“. -1.Aufl. New York: Garland Science, Taylor & Francis Group, 2008, S.5-8

Internetquellen

- [URL 1] <http://www.biotype.de/de/das-unternehmen/ueber-uns.html>, verfügbar am 10.08.2011
- [URL 2] <http://www.infoescola.com/wp-content/uploads/2011/03/PCR.jpg>, verfügbar am 10.08.2011
- [URL 3] [http://infohost.nmt.edu/~klathrp/dna-base pairs.gif](http://infohost.nmt.edu/~klathrp/dna-base%20pairs.gif), verfügbar am 25.05.2011
- [URL 4] <http://www.basic.northwestern.edu/biotools/oligocalc.html>, verfügbar am 15.08.2011
- [URL 5] http://www.eshca.net/java/books/javainsel8/javainsel_04_006.htm, verfügbar am 15.08.2011
- [URL 6] <http://www.ebi.ac.uk/help/matrix.html#dna>, verfügbar am 06.08.2011

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittwoch, den 29.08.2011

Janine Brettschneider